إصدار مكتبة الشروق

الذكاء الإصطناعى

Artifitial-Intelligence

من الأساسيات وحتى النهايات

نسخة عربى

الفهرس

الفصل الأول: مدخل إلى الذكاء الصنعي 1
1- الذكاء 1
2- الذكاء الصنعي 1
3- مجالات الذكاء الصنعي 2
4- ما هو الحاسوب الذكي 3
5- مقارنات عامة 4
1- ما الفرق بين الذكاء الطبيعي والذكاء الصنعي؟ 4
2- ما الفرق بين المعالجة الحاسوبية والمعالجة البشرية للمعطيات؟ 4
3- ما الفرق بين الذكاء الصنعي والبرمجة التقليدية؟ 5
6- هندسة المعرفة 5
1- نطاق المعرفة 6
2- ما هي هندسة المعرفة؟ 6
3- طرق هندسة المعرفة 6
7- فلسفة تمثيل المعرفة 7
1- الدور الأول: إيجاد البديل عن المعرفة بهدف المحاكمة عليها. 8
2- الدور الثاني: إبداع مفاهيم عامة في المعرفة Ontologies 8
3- الدور الثالث: اعتماد نظرية تجزيئية للمحاكمة الذكية. 9
4- الدور الرابع: إيجاد وسط ليعبر الناس عن معرفتهم. 11
5- الدور الخامس: إيجاد وسيط ليعبر الناس عن معرفتهم 11
8- مراجع البحث 11

الفصل الثاني: حل المشاكل باستخدام الفرضيات وحساب الإسناديات 12
1- استخدام المنطق في المحاكمة 12
2- حساب الفرضيات 13
1- اللغة 13
2- قواعد الاستدلال 14
3- الدلالة 15
4- الحل 18
5- الحل بالنقض 19
6- عبارات هورن 20
3- حساب الإسناديات Predicates 21
1- اللغة 21
2- قواعد الاستدلال 22

3- التوحيد 22
4- الحل 25
5- استخراج الجواب 26
4- تمارين 27
5- مراجع البحث 28

الفصل الثالث: طرق أخرى لتمثيل المعرفة والنظم الخبيرة 29
1- نظم قواعد الإنتاج 29
1- السلسلة الأمامية 29
2- السلسلة الخلفية 30
3- مقارنة بين السلسلة الأمامية والسلسلة الخلفية 30
4- مشاكل قواعد الإنتاج وحلول 31
2- شبكات الدلالة 32
3- ترابط المفاهيم 38
4- الأطر Frames 41
5- السيناريو 46
6- تمارين 49
7- مراجع البحث 50

الفصل الرابع: إيجاد الحلول الموضعية وخوارزميات البحث 51
1- فضاء الحالات 51
2- البحث في فضاء الحالات 55
3- بيان المسائل الجزئية 59
4- الشجرة and-or 60
5- البحث التجريبي 61
6- مراجع البحث 63

الفصل الخامس: المعرفة والتفكير المتلبسين 64
1- مراجعة الاحتمالات 64
2- قاعدة بايز والمحاكمة باستخدامها 67
3- الشك 70
4- عوامل اليقين 71
5- المنطق العائم 72
6- مراجع البحث 74

الفصل السادس: التعلم 75
1- مقدمة 75
2- التعلم الاستقرائي: الشبكات العصبونية 76
3- التعلم الاستنتاجي عبر الأمثلة 79
4- الألعاب.....84
5- مراجع البحث 86
الفصل السابع: التواصل والإدراك والفعل 78
1- مقدمة 87
2- حلقة التحسس/ التخطيط/الفعل 87
3- تعلم دوال تجريبية 88
4- الجوائز عوضاً عن الأهداف 91
5- التخطيط في الألعاب ذات اللاعبين 92
6- إجراء min-Max 93
7- مراجع البحث 98
الفصل الثامن: استشراف المستقبل: الخوارزميات المتطورة 99
1- مقدمة 99
2- الخوارزميات الجينية 99
3- البرمجة الجينية 102
4- نظم ذكية هجينة 103
5- مراجع البحث 104

الفصل الأول: مدخل إلى الذكاء الصنعي

1. الذكاء

يتطلبُ فَهمُ الذكاء، باعتقادنا، فَهم كيفية اكتساب المعارف، وتمثيلها، وتخزينها، وكيف يجري توليد السلوك الذكي وتعلمه؛ كيف تتطور الدوافع والعواطف والأفضليات وكيف تُستخدم، وكيف تُحول الإشارات الحسية إلى رموز؛ وكيف تُعالَج الرموز لإجراء العمليات المنطقية، ومحاكمة ما جرى في الماضي، والتخطيط للمستقبل؛ وكيف تولد آلياتُ الذكاء ظاهرةَ الوهم، والاعتقاد، والأمل، والخوف، والأحلام – حتى اللطف والحب. وباعتقادنا، أن تفهم الأساس الذي تقوم عليه هذه الوظائف، سيكون إنجازاً علمياً من مستوى الفيزياء النووية ونظرية النسبية وعلم الوراثة الجزيئية.

- **الذكاء:** هو القدرة على الفهم وتعلم الأشياء.
- **الذكاء (في القاموس):** هو القدرة على التفكير والفهم عوضاً عن القيام بالأعمال بالفطرة أو آلياً.
- **الذكاء الطبيعي:** كائن له دماغ، أو شيء ما، يمكنّه من التعلم، والفهم، وحل المشاكل واتخاذ القرارات.

2. الذكاء الصنعي

علم يبحث في السلوك الذكي لغير الكائنات الحية.

يستلزم السلوك الذكي بدوره الإدراك **perception**، والمحاكمة **reasoning**، و التعلم **learning**، والتواصل **communicating**، والتصرف **acting** في بيئات معقدة.

أحد أهداف الذكاء الصنعي الطويلة الأمد هو إنشاء آلات يمكنها فعل هذه الأشياء بالجودة التي يقدمها البشر، أو حتى على نحوٍ أفضل من البشر. الغرض الآخر من الذكاء الصنعي هو فهم هذا النوع من السلوك سواء حدث من الآلات أم من البشر أم من حيوانات أخرى. ومن ثَم، يكون للذكاء الصنعي أهداف علمية وهندسية في آنٍ واحد.

يعتقد البعض أن الآلات المفكرة قد يجب أن تكون معقدة جداً، وأن تمتلك خبرات معقدة (مثل التفاعل مع بيئاتها ومع آلات مفكرة أخرى) بحيث لا يمكننا فعلياً تصميمها أو بناؤها. وتعطي الإجرائيات التي تولد الطقس العام مثالاً جيداً لذلك. فحتى لو علمنا كل ما هو هام عن الطقس فلن نستطيع بالضرورة أن نصنع نسخة مطابقة لِهذه الظاهرة بكل غناها. لا يوجد نظام يستطيع صنع نسخة مطابقة لِظاهرة الطقس بجميع تفاصيلها ويكون تعقيده أقلّ من سطح الأرض الفعلي ، وغلافها الجوي وبحارها ، الموجودة في الفضاء، وتدفئة الشمس لها، وتأثير المد والجزر فيها.

بالمشابهة، سيكون بناء ذكاء من مستوى ذكاء البشر على جميع الأصعدة، أمراً معقداً جداً، أو على الأقل معتمداً بشدة على الفيزيولوجيا الدقيقة للبشر، بحيث يتعذّر أن يوجد بصورة مستقلة عن تَجسده في بشرٍ موجودين في بيئتهم.

لم يُبت بعدُ جواب السؤال الآتي:

هل سنتمكن فعلاً من بناء آلات مفكرة على مستوى البشر؟

وقد كان تقدم الذكاء الصنعي نحو هذا الهدف ثابتاً، وإن كان أبطأ مما توقع الرواد الأوائل. ونحن متفائلون فيما يتعلق بالنجاح النهائي.

أما تعريف الذكاء الصنعي، فهو يمكن أن يأخذ أحد الأشكال التالية، فيما يلي تعريفات اعتمدت له:

الذكاء الصنعي:

"جهود مثيرة لجعل الحواسيب (الآلات) تفكر.. أن يكون لها عقل بمعنى الكلمة"

(Haugeland 1985)

دراسة الإمكانات الذهنية عبر استخدام نماذج حسابية

(Charniak & McDermott 1985)

جزء من علوم الحاسوب يهتم بتصميم نظم حاسوبية ذكية تتصف بالخواص التي نصف بها البشر الأذكياء: فهم اللغات، التعلم، المحاكمة، حل المسائل، وغيرها...

(Barr & Feigenbaum (Jackson chapter 2))

3. مجالات الذكاء الصنعي

للذكاء الصنعي مجالات كثيرة ومتعددة، يبين الشكل أهمها، وتتضمن المجالات التالية:

- فهم اللغات الطبيعية المكتوبة ثم المنطوقة.
- تعرف الكلام وفهمه وتوليده حاسوبياً.
- الرؤية الحاسوبية وتعرف المشاهد واتخاذ قرارات.
- نظم التعليم الذكية بمعونة الحاسوب.
- الشبكات العصبونية.
- الربوطية ونظم التحـسس (رؤية وحركية ومعالجة إشارة للقيام بأفعال واتخاذ قرارات).
- الألعاب.
- برمجيات الترجمة وتلخيص المعلومات.
- البرمجة الآلية (برامج ذكية تنشئ أخرى)....

4. ما هو الحاسوب الذكي؟

- هل هو حاسوب لديه معطيات **Data**؟ أي لديه أرقام ومحارف وأسماء خام..؟ بالطبع لا، فتجميع المعطيات وخزنها ليس فيه كبير ذكاء.
- هل هو حاسوب لديه معلومات **Information**؟ أي إنه يتضمن نتائج معالجة رياضية وإحصائية لإعطاء توصيف مفيد للمعطيات، مثل إيجاد متوسط جميع القيم أو إيجاد القيمة الأعلى تكراراً أو القيمة الدنيا،..؟ أيضاً لا، إذ لا تتطلب مثل هذه المعالجات أي ذكاء، وهي مجرد عمليات حسابية على المعطيات.
- هل هو نظام معلومات يشمل كيفية تخزين واسترجاع المعلومات؟ هنا أيضاً نتحدث عن قواعد معطيات فقط، وإن كان التوجه حالياً إلى نظم قواعد معطيات ذكية.
- هل هو نظام خبير؟ يجمع المعطيات، ينشئ معلومات، يستشير خبيراً عن صلاحية المعلومات والقواعد عليها، يضع برنامجاً لحل المسائل المتعلقة بهذه الخبرة؟ هنا فقط بدأنا نتحدث عن خبرة ومعرفة وذكاء.

بإمكاننا سرد عدة وجهات نظر حول أنماط الآلات التي يمكن أن تكون قادرة على التفكير كالبشر . ومن الادعاءات التي تُسمع غالباً:

- إن الدماغ يعالج المعلومات على التوازي، في حين تقوم الحواسيب التقليدية بذلك تسلسلياً. ومن ثَم علينا بناء مجموعة منوعة جديدة من الحواسيب المتوازية لتحقيق تقدم في الذكاء الصنعي.
- تعتمد آلية الحساب التقليدية على المنطق الاثناني **true** أو **false**. وعلى النظم الذكية فعلاً أن تستخدم نوعاً من "المنطق الترجيحي **fuzzy logic**".
- تعتبر العصبونات الحيوانية أكثر تعقيداً من المباديل **Switches** - وهي كتل البناء الأساسية للحواسيب. لذا سنكون بحاجة إلى استخدام عصبونات صنعية حقيقية في الآلات الذكية.

اقترح Turing اختباراً، سمي اختبار Turing، يمكّننا من الحكم على آلة معينة: أهي ذكية أم لا. في البداية، جرى وصف الاختبار كأنه لعبة. يشارك في اللعبة ثلاثة أشخاص، رجلٌ (A) وامرأة (B) ومستجوب Interrogator (C) يمكن أن يكون من أحد الجنسين. يبقى المستجوب في غرفة بعيداً عن الاثنين الآخرين (يتصل معهما بواسطة الكتابة من بُعد).

هدف اللعبة: أن يعين المستجوب أي من الشخصين الآخرين هو الرجل وأيهما المرأة. هو يعرفهما بلصيقتين **X** و**Y**، وفي نهاية اللعبة يقول إما "**X** هو **A** و**Y** هو **B**" أو "**X** هو **B** و**Y** هو **A**".

يُسمح للمستجوب أن يطرح على **A** و**B** أسئلة مثل:

C: أرجو من **X** إخباري طول شعره (أو شعرها) ؟

نفترض الآن أن **X** هو فعلاً **A**، يجب على **A** عندئذ أن يجيب. ودور **A** في هذه اللعبة أن يضلل **C** ليجعله يخطئ في التعرف.

ماذا يجري حين يكون أحد اللاعبين **A** و**B** آلة؟ غالباً ما يُبسط اختبار **Turing** إلى اختبار تحاول فيه الآلة إقناع مستجوب بشري أنها بشر.

جرى تقديم عدة إصدارات من هذا الاختبار (**Turing**) البسيط، ونظراً لأن بإمكان آلات بسيطة جداً أن تخدع المستجوب البشري مُدة، فلا يُعد الإصدار البسيط عادةً اختباراً فعالاً لذكاء آلة ما. على سبيل المثال، يستخدمُ برنامجُ **ELISA** الذي ابتكره **Joseph Weizenbaum** بعض الحيل البسيطة جداً (كما في المقتطفات الذي يظهرها الإيضاح)، ولكنه يبدو للمستخدم المتساهل قادراً على إجراء حوارٍ واقعي بعض الشيء، حتى لو كان لا معنى له. أما برنامج **JULIA** الذي ابتكره **Mauldin** فهو برنامج حوارٍ أكثر حداثةً وتعقيداً.

5. مقارنات هامة

ما الفرق بين الذكاء الطبيعي والذكاء الصنعي؟

الذكاء الصنعي دائم بمعنى أنه متى حصلنا على نظام خبير برمجي فهو يبقى لدينا، في حين لا نستفيد من خبرة وذكاء البشر إلا أثناء وجودهم بيننا.

الحصول على نسخ مكررة من النظام البرمجي الذكي ممكن وسهل المنال، في حين لا يمكن بسهولة نقل خبرة الخبير البشري إلى شخص آخر لنحصل على نسخة جديدة من خبرة الخبير.

يتطلب الحصول على الخبير البشري، في غالب الأحيان، تكلفة أعلى بكثير من الحصول على البرنامج الذكي.

الذكاء الصنعي متسق، بمعنى أننا في غالب الأحيان نستطيع أن نعلم على ماذا اعتمد البرنامج الذكي في اتخاذ قراراته، أما القرارات البشرية فلا نستطيع تفسيرها تفسيراً متسقاً ولا نستطيع التنبؤ بها . ولكن بالمقابل، الذكاء البشري خلاق ومبدع أما الصنعي فهو نمطي وموثق وليس فيه مفاجآت.

ما لفرق بين المعالجة الحاسوبية والمعالجة البشرية للمعطيات؟

مع أن المعالجة الحاسوبية هي أصلاً منتج بشري، إلا أن المعالجة البشرية غير الحاسوبية تختلف عن المعالجة الحاسوبية للمسائل، والجدول التالي يبين بعض هذه الفروق:

المعالجة البشرية	المعالجة الصنعية (الحاسوبية)

تستخدم رموزاً لتمثيل مفاهيم المسألة ومعالجتها بهدف اتخاذ قرارات	تستخدم أرقاماً، ومحارف
ثمة تمثيل مجرد للمعرفة[1]، وجداول[2] يجري تعلمها لحل المسائل	تستخدم الذاكرة والملفات لتخزين المعلومات والبرامج.
تستخدم برمجة منطقية وغيرها	تستخدم خوارزميات
تستخدم الكسبيات (التجريبيات Heuristics) والاستدلال ومواءمة الأشكال لاتخاذ قرارات أعقد ناجمة عن الخبرة	تنتج قرارات بسيطة...

ما الفرق بين الذكاء الصنعي والبرمجة التقليدية؟

المقصود هنا هو الفرق بين تقنيات الذكاء الصنعي وبرمجتها (استخدام لغات الذكاء الصنعي مثل **prolog، lisp، C++** الغرضية التوجه، **clips، Jess** وغيرها) والبرمجة التقليدية بلغات إجرائية (**C، Pascal، C#،...**). إننا لا ننسى أن تنفيذ لغات الذكاء الصنعي يستخدم لغات إجرائية، وبالنهاية يجري العمل مع الحواسيب التي تتعامل أصلاً مع بتات فقط نستخدمها لتمثيل مختلف المعطيات، إلا أن من وضع لغات الذكاء الصنعي وضع نصب عينيه فلسفة عمل مختلفة يبينها الجدول التالي:

الذكاء الصنعي	البرمجة التقليدية
يقوم بمعالجة رموز	تقوم بمعالجة حسابية
قد لا يكون الدخل والخرج معروفين تماماً في النظم الذكية	الدخل والخرج معرفان تماماً بالخوارزميات
البحث عن الحل تجريبي	ثمة خوارزميات للبحث
التركيز على المعرفة	التركيز على المعطيات والمعلومات
فصل التحكم عن المعطيات، إضافة معطيات جديدة مستقلة عن إضافة أدوات محاكمة وتحكم جديدة	المعطيات مدمجة مع أدوات التحكم، وأي تغيير يتطلب إعادة ترجمة البرامج قبل التنفيذ
سهلة التحديث نسبياً	صعبة التحديث للسبب السابق

6. هندسة المعرفة Knowledge Engineering

إن المعرفة هي المادة الأولية للذكاء الصنعي، وهي أعلى من المعطيات والمعلومات، إذ تتضمن الخبرة في مجال معين. على سبيل المثال، تقضي الخبرة في مجال كهرباء السيارات ، بفحص صلاحية بطارية السيارة أولاً حين لا يُدار المحرك.

[1] أصوات، صور، رموز،...

[2] مثال: جداول الضرب، طريقة القسمة،...

نطاق المعرفة

هي معرفة الخبير التي يستخدمها لحل مشاكل في هذا النطاق. غالباً يكون نطاق المعرفة جزءاً من نطاق المسألة، وقد لا يعرف الخبير كل شيء في نطاق المسألة. قد تكون الخبرة سطحية وقد تكون عميقة.

- **المعرفة السطحية:**

تتكون من قواعد تجريبية تصف الربط بين مختلف الأغراض في نطاق المعرفة، وهي عملياً ما يطبقه الخبير في معظم الأحيان. إنها معرفة تجريبية وغير مؤكدة دائماً.

- **المعرفة العميقة:**

وهي تتعلق بالمبادئ والقوانين الأساسية التي تتحكم في النطاق، ففي المثال الذي ذكرناه أعلاه عن الخبرة في كهرباء السيارات، المعرفة العميقة تتطلب معرفة السبب الذي يجعل البطارية مسؤولة عن العطل السابق.

ليست المعرفة العميقة أفضل من السطحية، ولكل منهما أهميته. يستخدم الخبير المعرفة السطحية للمسائل العامة، والمعرفة العميقة للمسائل الأعقد حيث لا تكفي المعرفة السطحية، أو حين لا تكون المعرفة العميقة أصلاً موجودة. لاسيما وأن الحصول على المعرفة العميقة أصعب، وكذلك تمثيلها.

ما هي هندسة المعرفة؟

- هي تحصيل المعرفة في مجال ما، من مصدر أو أكثر غير إلكتروني، وتحويلها إلى شكل يمكن استخدامه في الحاسوب، لحل مسائل يجري حلها عادة من قبل أشخاص ذوي معرفة كثيفة في هذا المجال. وقد بُذلت جهود كبيرة لتطوير كميات كبيرة من المعرفة الفعالة.
- يمكن ألا تكون المسألة التي نريد حلها معرفة تماماً. فقد لا نستطيع تصميم حل منذ البداية، وقد نضطر لتغيير تمثيل المعرفة مع تغير مهام المسألة.
- قد لا يكون للمسائل حل واضح لفحص النظام، ونتساءل هل توجد حلول معيارية؟
- يمكن أن يكون مجال المسألة معقداً جداً، ويتطلب تجميع كم كبير من المعرفة، و هذه المعرفة هي في الغالب غير دقيقة، وغير معرفة تماماً، وغير مفهومة جيداً. ويصعب في الغالب تحويلها إلى شكل حاسوبي.

طرائق هندسة المعرفة

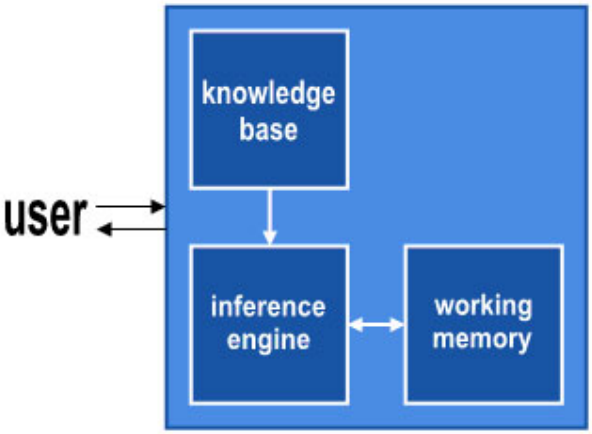
- **تحقيق نمذجة سريعة:** وذلك
  - بتطوير سريع لنموذج عمل **prototype**.
  - والتعلم من الأخطاء في بناء النماذج اللاحقة
  - والقيام بمراجعة دورية للنظام ولقاعدة المعرفة بناء على الاختبارات التي نجريها على النظام

➢ **اتباع منهجية مهيكلة:**

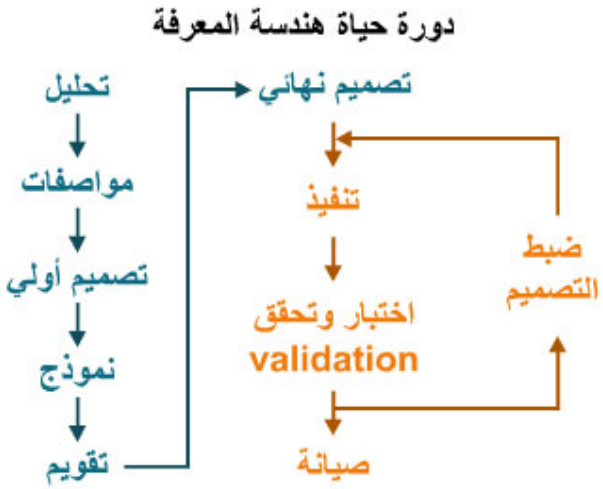
بحيث نعرف في كل لحظة، ما قمنا به، وأين نحن من هدفنا، وكيفية تحقيق الهدف. وبهذا تكون دورة حياة هندسة المعرفة كما في الإيضاح.

وبالنتيجة:

- صيانة وتطور نظم قواعد المعرفة نشاط مفتوح لا ينتهي.
- النظم الخبيرة لا تعتمد على الخوارزميات وإنما على المعرفة. قد تتغير المعرفة القديمة ، و قد نحصل معارف جديدة، إذن أداء النظم الخبيرة في تحسن مستمر.
- تشير نظم قواعد المعرفة في غالب الأحيان إلى النظم الخبيرة. والنظام الخبير هو اجتماع قاعدة معرفة **knowledge base**، ومحرك استدلال **inference engine** يستخدم هذه المعارف للمحاكمة عليها ولاستنتاج معارف جديدة. يُضاف إلى هذين المكونين واجهة تخاطبية للتفاعل مع المستخدم خبيراً كان أم مستثمراً ، وذاكرة عمل **Working Memory** مشتركة لجميع المكونات.



مكونات نظام خبير (نظم قواعد المعرفة)



7. فلسفة تمثيل المعرفة

حين نود تمثيل المعرفة، علينا تذكر الأدوار الخمسة التي يمكن أن يلعبها التمثيل وهي:

1. إيجاد بديل عن المعرفة نفسها، يقوم مقامها، ويساعد كياناً معيناً على المحاكمة وتحديد النتائج

اعتماداً على التفكير عوضاً عن القيام بأفعال.

2. إبداع مفاهيم عامة في المعرفة **Ontologies** (كيف يجب أن ننظر إلى العالم؟).
3. اعتماد نظرية تجزيئية للمحاكمة الذكية (كيفية المحاكمة، معرفة ما يقننه (يجعله قانونياً) الاستدلال وما ينصح به).
4. إيجاد وسط للحسابات الفعالة: كيف يقود التمثيل إلى تنظيم المعلومات لتسهيل الاستدلال المنصوح به.
5. إيجاد وسط ليعبر الناس عن معرفتهم: لغة ليقولوا أشياء عن العالم حولهم.

كل من هذه الأدوار يقود إلى خواص هامة وربما مختلفة لتمثيل المعرفة. ويحدد كل من هذه الأدوار إطار عمل قد يكون مختلفاً من دور إلى آخر. يجب تحديد ما هو مهم، لالتقاط وتمثيل الغنى الموجود في العالم الطبيعي حولنا بغية تمثيله والمحاكمة عليه والقيام بأفعال مناسبة.

الدور الأول: إيجاد البديل عن المعرفة بهدف المحاكمة عليها

إن المحاكمة على المعرفة شيء داخلي يُطبق على أشياء خارجية. والسؤال الأول الذي يطرح نفسه هو:

"ما دلالة هذه المعرفة؟"

وثمة سؤال هام آخر هو:

"ما دقة تمثيل المعرفة وما قرب هذا التمثيل من الحقيقة؟"

لاسيما وأننا يمكن أن نمثل أشياء مادية (مثل عجلة دراجة، مفتاح، منضدة...)، أو أشياء مجردة (مثل أفعال، إجرائيات، اعتقاد...)

بما أن البدائل ليست دقيقة تماماً، أي أن الغرض لن يتساوى مع تمثيله، فمن المؤكد أن المحاكمة باستخدام التمثيل قد لا تكون صحيحة تماماً. في تمثيل المعرفة، نحذف أشياء ونضطر إلى إضافة أشياء أخرى. نود أن يقلل التمثيل ما أمكن من أخطاء المحاكمة باستخدام التمثيل.

على سبيل المثال: حين نعرف المنضدة على أنها شيء له شكل معين، نغفل أبعادها فقد تشترك عدة طاولات في الشكل وتكون أبعادها مختلفة (إحداها مصغّرة عن أخرى..)، ونغفل المادة المصنوعة منها...

الدور الثاني: إبداع مفاهيم عامة في المعرفة

بما أننا لسنا دقيقين في التمثيل، فلابد من اتخاذ قرارات حول ما يجب علينا رؤيته من العالم حولنا (المنضدة: هل هي سطح مستو؟ هل يمكن أن تحمل أشياء أخرى؟ هل تشكل عقبة في الطريق؟ هل لها شكل محدد؟...)؛ وأن نركّز على ما نعتقد أنه مهم ولازم للمهام المطلوبة منها.

حين نود أن نحاكم حول إمكان تحرك ربوط في ممر يوجد فيه منضدة، فما يهمنا هو أبعاد المنـضدة وإن كان وجودها يعيق حركة الربوط. وحين نحاكم حول إمكان اندلاع حريق في مكان المنضدة، يهمنا معرفة المادة المصنوعة منها. وحين نحاكم حول إمكان وضع مكعبات على المنضدة تهمنا وضعيتها ومدى ملاسة سطحها...

يمكن كتابة هذه المفاهيم بعدة لغات برمجة (مثل **Lisp, Prolog, CLIPS**...)، لا يهم بأي لغة؛ المهم هو ما تعنيه هذه المفاهيم.

اعتماداً على ما سبق قد نختار المنطق لتمثيل المعرفة، أو الأطر **Frames** أو قواعد الإنتاج (قواعد شرطية من الشكل **if, then**)، يجب اختيار طريقة التمثيل منذ البداية. يمكن اختيار أكثر من طريقة تمثيل للمشكلة نفسها. على سبيل المثال، في مسألة تشخيص الأمراض اعتمد النظام الخبير **MYCIN** قواعد الإنتاج، في حين اعتمد النظام الخبير **INTERNIST** على الأطر فعرف نماذج أولية لأمراض الكلى أو أمراض الكبد.

الدور الثالث: نظرية تجزيئية للمحاكمة الذكية

يتضمن التمثيل جزءاً من الاعتقاد أو النظرة التي دفعت إليه. وهذه النظرة هي جزء من الظاهرة المعقدة المتعددة الوجوه للمحاكمة الذكية.

مكونات هذه النظرة هي:

- المفهوم الأساسي لتمثيل الاستدلال الذكي،
- معرفة الاستدلال الذي يقننه التمثيل،
- ومعرفة الاستدلال الذي ينصح به التمثيل.

وهذه المكونات هي الجواب على الأسئلة الثلاثة التالية:

- **ما هي المحاكمة الذكية؟**

- **رياضياً:** هي التي تعتمد المنطق والاستنتاج، ويكون الحل اعتماداً على الفرضيات ، ثم الإسناديات، ثم لغة البرمجة المنطقية **PROLOG**.
- **نفسياً:** هي التي تقلد سلوك الإنسان (مثل حل المسائل بالتخطيط والوصول إلى الهدف ، والنظم الخبيرة).
- **حيوياً:** تصرف يكون استجابة لتحريض، اعتماداً على ربط متوازٍ لعدة معالجات بسيطة (مثل الشبكات العصبونية الصنعية **ANN**، والوكلاء **agents**).
- **إحصائياً:** هي الخضوع إلى قوانين الاحتمالات (مثل المحاكمة على معارف غير مؤكدة).
- **اقتصادياً:** هي نظرية المنفعة.

وقد شهد التاريخ **عظماء كثيرين** كانوا وراء استخدام الأوجه المختلفة للمحاكمة الذكية.

Mathematical	Psychology	Biology	Statistics	Economics
Logic				
Aristotle				
Descartes				
Boole	James		Laplace	Bentham, Pareto
Frege		Bernoulli	Friedman	
Peano				
	Hebb	Lashley	Bayes	
Goedel	Bruner	Rosenblatt		
Post	Miller	Ashby	Tversky	Von Neumann
Church	Newell	Lettvin	Kahneman	Simon
Turing	Simon	McCulloch, Pitts		Raiffa
Davis		Heubel, Weisel		
Putnam				
Robinson				
LOGIC	SOAR	CONNECTIONIS M	Causal networks	Rational agents
PROLOG	Knowledge-based systems	A-life		
	Frames			

Views of Intelligent Reasoning and Their Intellectual Origins

الدور الثالث: نظرية تجزيئية للمحاكمة الذكية

➢ الاستدلال الذي يقننه التمثيل؟ (ماذا يمكن أن نستنتج؟)

إنها **الاستنتاجات السليمة Soundness** مثل قاعدة مودس بونتز : الموت نهاية كل إنسان، وسقراط إنسان، إذا نستنتج أن نهاية سقراط الموت، ومع ذلك فنحن نستخدم أيضاً **الاستقراء Abduction** إذا كانت الحمى التيفية تؤدي إلى ارتفاع الحرارة، فربما نستقرئ الحمى التيفية من ارتفاع الحرارة - رغم وجود أسباب أخرى عديدة لارتفاع الحرارة. بعد المنطق نشأت الأطر التي تعطي استنتاجات قد تكون خاطئة ولكنها تعطي دائماً طباقاً واضحاً **matching**.

➢ الاستدلال الذي ينصح به التمثيل؟ (ماذا يجب أن نستنتج؟)

إن ما يمكن أن نستنتجه كثير جداً، فما هو المهم بين هذا الكم الكثير من الاستنتاجات؟

المنطق لا يعطي الاستنتاج المنصوح به. على سبيل المثال، باستخدام قاعدة مودس بونتز على قاعدة معرفة تضم حقائق، نحصل على مجموعة حقائق جديدة، قد لا نكون بحاجة إليها في الوصول إلى أهدافنا أو محاكمتنا.

لذلك نفكر في طرق جديدة لتمثيل المعرفة مثل الأطر **frames**، والنماذج المعينة للتصرف **stereotypes, scripts**. (نصف ماذا نفعل في حفل عيد ميلاد؟ في المطعم؟ في الحافلة؟...)

إنها التقاط المعارف البديهية البسيطة التي تصعب برمجتها!

ربما كان على المستخدم أن يخبر النظام ماذا عليه أن يفعل، أو أن يقوده في فعل ما هو صحيح، أو أن يبني إستراتيجية استدلال خاصة به.

الدور الرابع: إيجاد وسط للحسابات الفعالة

في النهاية علينا تنفيذ تمثيل المعرفة والمحاكمة عليها باستخدام حواسيب وآلات و عمليات حسابية ، فكيف ننظم المعلومات لتسهيل الحصول على الاستدلال الأفضل على هذه الحواسيب؟

الهدف من استخدام الأطر كان هذا:

نُرفق بكل إطار عدة أنواع من المعلومات، بعضها حول كيفية استخدام الإطار، وبعضها حول ما نتوقع أن يحدث لا حقاً، وبعضها عما يجب فعله إذا لم تصدق هذه التوقعات... لذلك نصنف الأطر بأنها تسمح بالحصول على محاكمات تصنيفية **Taxonomic**. وتسير شبكات الدلالة باتجاه إيجاد روابط بسيطة بين العقد. أما نظم قواعد المعرفة فتنطلق من الهدف (سلسلة خلفية) أو من الحقائق (سلسلة أمامية) لإيجاد المعارف اللازمة المفيدة.

يستبعد علماء مضمون المعرفة **epistemology** عموماً الاهتمام بالحسابات.

الدور الخامس: إيجاد وسط ليعبر الناس عن معرفتهم

العالم حولنا غني بالمعارف، كيف نخبر الحواسيب بمعرفتنا عن العالم، أو كيف نخبر بها شخصاً آخر على أقل تقدير؟

ما مدى ملاءمة تمثيل المعرفة لوسط التعبير؟ ما مدى عموميته؟ وما مدى دقته؟... ما مدى ملاءمته للتواصل في هذا الوسط؟ وللتعامل مع لغته؟ ما هي الأشياء التي يسهل التعبير عنها في لغته؟ وما هي الأمور التي يصعب أو يستحيل التعبير عنها في هذه اللغة؟

بناء على ما تقدم نستنتج النقاط التالية:

- تمثيل المعرفة هو اللغة التي يجب أن نعبر بها ونتواصل بها. لذلك يجب أن نتكلمها دون كبير عناء.
- تمثيل المعرفة هو إطار عمل يساعد على إيجاد وجهات نظر صريحة وهامة في التمثيل والبحث في روح التمثيل.
- يجب توضيح الفرق في التصميم الناتج عن التساهل مع هذه الروح عوضاً عن إجبارها وخرقها.
- ما هي الأفكار التي نود تمثيلها؟ كيف يمكن تنفيذها وباستخدام أية تقنية؟ وعلى أي حاسوب؟
- على الرغم من أهمية أدوات وتقنيات التمثيل، إلا إن فهم ووصف غنى العالم حولنا يجب أن يكون هو شغلنا الشاغل.

8. مراجع البحث

- "الذكاء الصنعي: رؤية جديدة" من منشورات الجمعية العلمية السورية للمعلوماتية.
- **Giarratano & Riley "Expert Systems: Principles and Programming", 3d Edition, 1998**
- **"What is Knowledge Representation" R. Davis et al. (on the web)**

الفصل الثاني: حل المشاكل باستخدام حساب الفرضيات وحساب الإسناديات

1. استخدام المنطق في المحاكمة

يمكن صياغة بعض المعلومات الصعبة التمثيل بصيغة قيود على قيم السمات. ويمكن في كثير من الأحيان، استخدام هذه القيود، التي تمثّل معارف هامة عن العالم حولنا، لاستنتاج قيم سمات أخرى لا يمكن تحسسها مباشرة. تختلف مسألة استنتاج المعلومات المتعلقة بالحالة الراهنة **present** (باستخدام حسابات تعتمد على القيود الموضوعة على السمات).عن مسألة حساب الحالات المستقبلية **future** (باستخدام طرائق البحث.)التي تنتج من جراء أفعال. نسمي العملية الأولى " **المحاكمة reasoning**" والثانية "**التخطيط planning**".

نركز الآن على مسألة المحاكمة.

هناك العديد من التطبيقات التي تتطلب محاكمةَ قيمِ السمات. فقد، ثَبتَ مثلاً أن بالإمكان تمثيل وظائف العديد من النظم الفيزيائية، ومنها النظم البيولوجية باستخدام مجموعة مناسبة من السمات . وترمز القيود الموضوعة على هذه السمات القوانين الفيزيائية، أو غيرها، المتعلقة بالكائن الحي أو بالجهاز . بعدئذ يمكن استخدام المحاكمة لأغراض أخرى كتشخيص الأعطال الوظيفية في هذه النظم ؛ إذ يمكن استنتاج السمات المرتبطة بـ "**الأسباب causes**" من السمات المرتبطة بـ " **الأعراض symptoms**". يعتبر هذا المنهج العام أساساً لطائفة هامة من تطبيقات الذكاء الصنعي، المعروفة بـ " **النظم الخبيرة expert systems**".

مثال:

ليكن لدينا ربوط قادر على رفْع كتلة: إذا كانت هذه الكتلة قابلة للحمل (أي غير ثقيلة جداً)، و كانت الشحنة الكهربائية المخزنة في بطارية الربوط كافية، عندها ستتحرك ذراع الربوط إذا ما حاول رفع الكتلة التي يمسكها. يمكن تمثيل مختلف هذه الشروط باستخدام السمات الاثنانية القيمة **(TRUE, FALSE)** التالية:

(BAT_OK) x_1
(LIFTABLE) x_2
(MOVES) x_3

لتسهيل النقاش سنطلق على هذه السمات أسماء (معطاة بين القوسين) يسهل تذكرها.

لنفترض أن الربوط يتحسس قيمة **BAT_OK** (بقراءة مقياس)، وقيمة **MOVES** (بطريق محسات موضوعة على مفاصله)، إلا أنه لا يتحسس قيمة **LIFTABLE**. علماً بأن قيمة **LIFTABLE** تعتبر هامة، وضرورية للربوط في بعض المهام الأخرى التي عليه إنجازها.

نعرف، من خلال هذا الوصف، أنه إذا كان لكل من **BAT_OK** و**LIFTABLE** القيمة **1** فستكون قيمة **MOVES** تساوي **1** أيضاً. لذلك، إذا كانت قيمة **MOVES** تساوي **0** عند محاولة الربوط تحريك الكتلة،

فإننا نعرف أن قيمة **BAT_OK** أو **LIFTABLE** (أو كليهما) تساوي **0**. لكن، إذا كانت قيمة **BAT_OK** وفقاً للمحس تساوي **1**، فلا بد أن تكون قيمة **LIFTABLE** مساوية **0**.

إذا كان بإمكاننا أن نُجري المحاكمة هكذا، فلْنجعل الربوط يقوم بذلك أيضاً . نحتاج إذن إلى لغة **language** يمكن بها التعبير عن القيود المفروضة على السمات وعن قيمها بآن واحد، إضافة إلى آلية استدلال **inference** تمكننا من تحقيق المحاكمة المطلوبة . يقدم حساب الفرضيات **propositional calculus**، وهو أحد فروع الجبر البولياني، الأدوات الضرورية لذلك.

يكتب القيد في المثال المطروح باستخدام لغة حساب الفرضيات كما يلي: **BAT_OK** ∧ **MOVES** ⊃ **LIFTABLE**

حيث ∧ تعني "و"، و⊃ تعني "يقتضي". ويمكن استخدام آليات مرافقة لِهذه اللغة، لاستخلاص نتائج من العبارات في اللغة.

- **يستلزم المنطق:**
  - لغة (لها تركيب نحوي لتحديد ما هو التعبير المسموح به في اللغة).
  - قواعد استدلال لمعالجة الجمل في اللغة.
  - دلالات لربط عناصر اللغة بعناصر بعض الموضوعات.

سندرس لغتين منطقيتين:

- أولاهما وأبسطهما وتسمى حساب الفرضيات **propositional calculus**؛
- وتسمى الثانية، وهي أكثر فائدة، حساب الإسناديات من المرتبة الأولى **first order predicate calculus (FOPC)**.

تسمح لغة حساب الفرضيات بعرض العديد من الأفكار الهامة المتعلقة بحساب الإسناديات من المرتبة الأولى بصورة أشد بساطة، لذلك سنعرض تلك اللغة أولاً.

2. حساب الفرضيات

اللغة

تتكون لغة حساب الفرضيات صورياً من المكونات التالية:

- **الذّرات atoms:** لدينا الذرتان T وF (أي صح **True** وخطأ **False**)، والمجموعة العدوْدة وغير المنتهية من سلاسل محرفية تبدأ بحرف كبير (مثل: **P, Q, R,..., P1, P2, ON_A_B,...** وهكذا).
- **الروابط connectives:** وهي ∨، ∧، ⊃، ¬، والتي تسمى على التتالي "أو"، "و"، "يقتضي"، "لا". سنعطي لاحقاً لهذه الروابط معنى يرتبط بأسمائها؛ أما الآن، فهي مجرد رموز غير ذات معنى.

يسمى التركيب النحوي للصيغ الجيدة التركيب **well-formed formulas** (**wffs**) جملاً:

- أية ذرة هي صيغة جيدة التركيب، مثال: **P**، **R** و**P3**.
- إذا كانت ω_1 و ω_2 صيغتين جيدتي التركيب فإن كلاً من الصيغ التالية هي كذلك:

صيغ جيدة التركيب باستخدام ω_1 و ω_2:

$\omega_1 \vee \omega_2$ (تسمى فصلاً لِـ ω_1 و ω_2)

$\omega_1 \wedge \omega_2$ (تسمى عطفاً من ω_1 و ω_2)

$\omega_1 \supset \omega_2$ (تسمى اقتضاء)

$\neg\omega_1$ (تسمى نفياً لِـ ω_1)

من الأمثلة على صيغ أخرى جيدة التركيب:

(P∧Q) ⊃ ¬P
P ⊃ ¬P
P∨ P ⊃ P
(P⊃Q) ⊃ (¬Q ⊃ ¬P)
¬¬P

- نسمي الذرات أو الذرات المسبوقة بإشارة ¬ حرفيات **Literal**. ونسمي ω_1 في $\omega_1 \supset \omega_2$ بمقدمة **antecedent** الاقتضاء و ω_2 نتيجةَ **consequent** الاقتضاء.
- ليس هناك صيغ أخرى جيدة التركيب، مثال: **P⊃¬¬** ليست صيغة جيدة التركيب.

تعتبر الأقواس في بعض المراجع جزءاً من اللغة، ولكننا سنستخدمها هنا بطريقة بديهية وغير دقيقة . يمكن تعرفُ الصيغ الجيدة التركيب باستخدام تعريفها العودي. فمثلاً، لبرهان أن **(P ∧ Q) ⊃ ¬ R** هي صيغة جيدة التركيب، نلاحظ أولاً أن **(P ∧ Q)** هي صيغة جيدة التركيب لأن **P** و**Q** كذلك. و**¬R** هي صيغة جيدة التركيب لأن **R** كذلك. ومن ثَم، فإن **(P ∧ Q) ⊃ ¬R** هي صيغة جيدة التركيب.

قواعد الاستدلال

ثمة عدة طرق لتوليد صيغ جيدة التركيب بدءاً من صيغ أخرى، نسمي ذلك قواعد الاستدلال **rules of inference**. وهي هنا مجرد طريقة ميكانيكية لإنشاء صيغة جيدة التركيب انطلاقاً من صيغ أخرى.

وبذلك، يكون لقاعدة الاستدلال نموذجياً الصيغةُ التالية:

يمكن استنتاج الصيغة γ من الصيغة α (أو من الصيغتين α وβ).

وفيما يلي مجموعة من قواعد الاستدلال الشائعة:

- يمكن استنتاج الصيغة ω_2 من الصيغتين ω_1 و$\omega_1 \supset \omega_2$ (تُسمى هذه القاعدة مودس بونس **modus ponens**).
- يمكن استنتاج الصيغة $\omega_1 \wedge \omega_2$ من الصيغتين ω_1 وω_2 (إضافة العطف $\wedge$).
- يمكن استنتاج الصيغة $\omega_2 \wedge \omega_1$ من الصيغة $\omega_1 \wedge \omega_2$ ($\wedge$ تبديلي).

- يمكن استنتاج الصيغة ω_1 من الصيغة $\omega_1 \wedge \omega_2$ (حذف العطف $\wedge$).
- يمكن استنتاج الصيغة $\omega_1 \vee \omega_2$ إما من ω_1 أو من ω_2 (إضافة الفصل $\vee$).
- يمكن استنتاج الصيغة ω_1 من الصيغة $(\neg\omega_1)$ $\neg$ (حذف النفي $\neg$).

تُسمى سلسلة الصيغ الجيدة التركيب $\{\omega_1, \omega_2, ..., \omega_n\}$ برهاناً **Proof** (أو استنتاجاً **deduction**) لِـ ω_n من مجموعة من الصيغ الجيدة التركيب Δ، إذا وفقط إذا وجدت كل صيغة ω_i ضمن Δ، أو كان بالإمكان استنتاجها من صيغة أو (عدة صيغ) سابقة في السلسلة، باستخدام إحدى قواعد الاستدلال. إذا كان هناك برهان لِـ ω_n من Δ، فإننا نقول إن ω_n هي مبرهنَة **theorem** للمجموعة Δ. ونستخدم هنا الترميز التالي للتعبير عن إمكان البرهنة على ω_n من Δ: $\Delta \models \omega_n$

مثال:

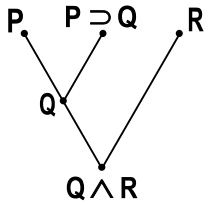
ليكن لدينا مجموعة الصيغ الجيدة التركيب Δ:

Δ **= {P, R, P** $\supset$ **Q}**

إن السلسلة التالية هي برهان لِـ **Q** $\wedge$ **R** باستخدام قواعد الاستدلال المعطاة في العبارة السابقة:

{P, P $\supset$ **Q, Q, R, Q** $\wedge$ **R}**

يمكن أن يعتمد مفهوم البرهان على ترتيب جزئي إضافة إلى التتالي. ويمكن تمثيل هذا الترتيب الجزئي ببنية شجرية. تُعنون كل عقدة في شجرة البرهان بصيغة **wff** التي يجب إما أن توافق إحدى صيغ Δ أو أن يكون بالإمكان استنتاجها من آبائها في الشجرة باستخدام إحدى قواعد الاستدلال. وتكون الشجرة المعنونة برهاناً للصيغة الموجودة في جذر الشجرة.



الدلالة

تقوم الدلالة **semantics** بالربط بين عناصر لغة منطقية وعناصر في مجال المعرفة. هذه الروابط هي ما نقصده بِـ"المعنى". نقوم في منطق الفرضيات (ومن هنا جاء اسم منطق الفرضيات) بربط الذرات بفرضيات **propositions** عن العالم.

فمثلاً، يمكن أن نربط الذرة **BAT_OK**[1] بالفرضية "البطارية مشحونة " **"The Battery is charged"**. تُسمى عملية ربط الذرات بالفرضيات بالتفسير **interpretation**. وفي تفسير معطى، تُسمى الفرضية المرتبطة بذرة رمز الذرة **denotation**.

يكون للذرات، ضمن تفسير معطى، القيمة **True** أو **False**. إذا كانت الذرة α مرتبطة بفرضية **P**،

[1] لسنا مضطرين إلى استخدام سلسلة محرفية سهلة التذكر، إذ يمكن الاستعاضة عنها بسلاسل محرفية أخرى.

نقول إن للذرة α القيمة **True** إذا كانت الفرضية **P** صحيحة في عالم المسألة، وإلاّ تكُنْ قيمتها **False**.

مثال: الذرة **Bat_Ok** تدل على فرضية كون البطارية بحالة جيدة، ولذا تأخذ هذه الذرة القيمة **True** حين تتحقق هذه الفرضية.إن قيمة الذرة الخاصة **T** هي دائماً **True**، وقيمة الذرة الخاصة **F** هي دائماً **False**. ولما كان من الضروري أن تكون فرضيات العالم إما صحيحة أو لا (وهذا غير واقعي دائماً!)، فبالإمكان تحديد قيمة تفسير بإسناد القيم مباشرة إلى الذرات في اللغة - بقطع النظر عن فرضية العالم التي ترمز إليها كل ذرة.

يعني الورود الصريح لعنصر ما (مثلاً **X1**) في قاعدة المعرفة أن الفرضية الموافقة صحيحة في عالمنا. سنرى لاحقاً كيفية استخدام الصيغ الجيدة التركيب في قاعدة المعرفة. يبين الجدول التالي دلالات الروابط في حساب الفرضيات:

ω_1	ω_2	$\omega_1 \wedge \omega_2$	$\omega_1 \vee \omega_2$	$\neg\omega_1$	$\omega_1 \supset \omega_2$
True	**True**	**True**	**True**	**False**	**True**
True	**False**	**False**	**True**	**False**	**False**
False	**True**	**False**	**True**	**True**	**True**
False	**False**	**False**	**False**	**True**	**True**

لنفترض أن **P** تأخذ القيمة **False**، و**Q** تأخذ القيمة **False** و**R** تأخذ القيمة **True**، وأننا نريد حساب قيمة الصيغة:

[(P ⊃ Q) ⊃ R] ⊃ p

بالحساب من الداخل باتجاه الخارج، نجد أولاً أن قيمة**Q ⊃ P** هي **True**؛ ثم نحسب قيمة **(P ⊃ Q) ⊃ R** فنجدها **True** أيضاً. ولما كانت **P** لها القيمة **False** كانت قيمة التعبير الكلي هي **False**.

نقول عن تفسير ما إنه يحقق صيغة جيدة التركيب إذا كانت قيمة هذه الصيغة وفق هذا التفسير هي **True**. ونقول عن تفسير يحقق صيغةً جيدة التركيب إنه نموذج **model** لها. ونقول عن تفسير يحقق كل صيغة جيدة التركيب في مجموعة من الصيغ الجيدة التركيب بأنه نموذج لهذه المجموعة.

يمكن ألا يكون هناك تفسير يحقق صيغة جيدة التركيب (أو مجموعة من الصيغ). نقول في هذه الحالة إن الصيغة (أو مجموعة الصيغ) غير متسقة أو غير قابلة للتحقيق. فالصيغتان **F** و**P ∧¬P** غير قابلتين للتحقيق (ليس هناك تفسير يجعل قيمة هاتين الصيغتين **True**). كذلك، فإن مجموعة الصيغ:

{P ∨ Q, P ∨¬Q, ¬P ∨ Q, ¬P ∨ ¬Q}

غير قابلة للتحقيق.

استخدم جدول الحقيقة ليتأكد لك أنه لا يوجد أي تفسير يجعل جميع صيغ هذه المجموعة صحيحة معاً.

نقول عن صيغة إنها صحيحة valid[2] إذا كانت قيمتها **True** في جميع تفاسير ذراتها المكونة. من الأمثلة على الصيغ الجيدة التركيب الصحيحة ما يلي:

$P \supset P$ ($\neg P \vee P$ وهي تكافئ)
T
$\neg (P \wedge \neg P)$
$Q \vee T$
$[(P \supset Q) \supset P] \supset P$
$P \supset (Q \supset P)$

نقول عن صيغتين جيدتي التركيب إنهما متكافئتان**equivalent** إذا وفقط إذا كانت لهما قيم الحقيقة نفسها في جميع التفاسير. انظر التكافؤات التالية:

قانونا دمورغان **DeMorgan**:

$\neg (\omega_1 \vee \omega_2) \equiv \neg\omega_1 \wedge \neg\omega_2$
$\neg (\omega_1 \wedge \omega_2) \equiv \neg\omega_1 \vee \neg\omega_2$

قانون عكس الإيجاب **Contrapositive**:

$(\omega_1 \supset \omega_2) \equiv (\neg\omega_2 \supset \neg\omega_1)$

إذا كانت ω_1 وω_2 متكافئتين فإن الصيغة التالية صحيحة:

$(\omega_1 \supset \omega_2) \wedge (\omega_2 \supset \omega_1)$

وبسبب هذه الحقيقة، كثيراً ما تُستخدم الكتابة $\omega_1 \equiv \omega_2$ على أنه اختصارٌ لـ$(\omega_2 \supset \omega_1) \wedge (\omega_1 \supset \omega_2)$.

إذا كان لصيغة جيدة التركيب القيمة **True** في كل التفسيرات التي تجعل لكل صيغة جيدة التركيب في مجموعة Δ من الصيغ القيمة **True**، فإننا نقول إن Δ تستلزم منطقياً ω، وإن ω تتبع منطقياً Δ أو إن ω هي نتيجة منطقية لـ Δ. نستخدم الرمز $\models$ للدلالة على الاستلزام المنطقي، ونكتب $\Delta \models \omega$. مثلاً:

$\{P\} \models P$
$\{P, P \supset Q\} \models Q$
$F \models \omega$

حيث ω هي أية صيغة جيدة التركيب

$P \wedge Q \models P$

يستخدم الاستلزام المنطقي كثيراً في مسائل الذكاء الصنعي، إذ يقدم طريقة فعالة لإظهار أنه إذا كانـت مجموعة فرضيات عن العالم صحيحة، فإن فرضية هامة أخرى (ربما لا يمكن تحسسها) يجب أن تكون صحيحة أيضاً.

إذا تحقق أنه حين تبرهن المجموعة Δ بحسب قواعد الاستدلال على الصيغة ω، فإن هذا يقتضي أن Δ تستلزم ω؛ فإننا نقول إن مجموعة قواعد الاستدلال سليمة.

[2] إن الصيغة الجيدة التركيب الصحيحة تكون عديمة الأهمية؛ لأنها لا تُعطي أية معلومة عن حالة العالم.

وإذا تحقق أنه متى كان $\omega \models \Delta$ يوجد برهان $\vdash \omega$ من Δ باستخدام مجموعة قواعد الاستدلال، فإننا نقول إن مجموعة القواعد كاملة.

عندما تكون قواعد الاستدلال سليمة وكاملة، فيمكن أن نُحدد وجود أو عدم وجود تبعية صيغة ما جيدة التركيب تَبعيةً منطقية لِمجموعة من الصيغ جيدة التركيب، وذلك بالبحث عن برهـان لهـذه الصيغة عوضاً عن استخدام جدول الحقيقة.

عندما تكون قواعد الاستدلال سليمة، ونجد برهاناً $\vdash \omega$ من Δ، فإننا نعرف أن ω تتبع منطقياً Δ. وعندما تكون قواعد الاستدلال كاملة، فإننا نعرف أننا سنكون قادرين على تأكيد أن ω تتبع Δ (عنـدما تكون كذلك فعلاً) باستخدام إجراء بحث كامل للبحث عن برهان. إن الاستعاضة عن طرائق جدول الحقيقة بطرائق البرهان يجعل تكلفة الحسابات في حساب القرضيات وحساب الإسناديات (الذي سندرسه لاحقاً) أقل بكثير.

الحل Resolution

لتكن Σ_1 وΣ_2 مجموعتين من الحرفيات وλ ذرة. يمكن من الصيغتين $\{\lambda\}\cup\Sigma_1$ و$\{\neg\lambda\}\cup\Sigma_2$ أن نستنتج أن $\Sigma_1\cup\Sigma_2$، وهذا ما يسمى ناتج حل العبارتين Σ_1 وΣ_2 (أو اختصاراً حلّ). وتُسمى الذرة λ المحلول ، والإجرائية الحل.

فيما يلي بعض الأمثلة:

- يؤدي حل $\mathbf{P} \vee \mathbf{R}$ و$\neg\mathbf{P} \vee \mathbf{Q}$ إلى $\mathbf{R} \vee \mathbf{Q}$. **كيف؟**
  يمكن أن نكتب العبارات التي جرى حلها كاقتضاءين $\neg\mathbf{R} \supset \mathbf{P}$ و$\mathbf{P} \supset \mathbf{Q}$. عند تطبيق قاعدة الاستدلال المسماة السلْسلة **chaining** على الاقتضاءين السابقين نحصل على $\neg\mathbf{R} \supset \mathbf{Q}$، التي تُكافئ الحل $\mathbf{R} \vee \mathbf{Q}$. ونرى، أن السلسلة هي حالة خاصة من الحل.
- يؤدي حل $\mathbf{R}$ و$\neg\mathbf{R} \vee \mathbf{P}$ إلى $\mathbf{P}$: إذ كانت العبارة الثانية تُكافئ $\mathbf{R} \supset \mathbf{P}$، نلاحظ أن مودس بوننس هي أيضاً حالة خاصة من الحل.
- يؤدي حل $\mathbf{P} \vee \mathbf{Q} \vee \mathbf{R} \vee \mathbf{S}$ مع $\neg\mathbf{P} \vee \mathbf{Q} \vee \mathbf{W}$ على $\mathbf{P}$ إلى $\mathbf{Q} \vee \mathbf{R} \vee \mathbf{S} \vee \mathbf{W}$: لاحظ أن $\mathbf{Q}$ تظهر مرة واحدة في عبارة الحل (والتي هي معرفة باعتبارها مجموعة أصلاً).
- يؤدي حل λ (وهو حرفي موجب) مع $\neg\lambda$ (وهو حرفي سالب) إلى العبارة الخالية. أي يمكننا من λ و$\neg\lambda$ استنتاج $\mathbf{F}$ لأن λ و$\neg\lambda$ متناقضان: إن أية مجموعة من الصيغ الجيدة التركيب تحوي λ و$\neg\lambda$ هي مجموعة غير قابلة للتحقيق.

يمكن تحويل أية صيغة في حساب القرضيات إلى عطف من العبارات مكافئ لها. ونقول عن الصيغة المكتوبة كعطف من العبارات إنها من شكل العطف النظامي (كما نقول عن صيغة مكتوبة كفصل من

حرفيات معطوفة إنها من شكل الفصل النظامي).

مثال

سنستعرض في المثال التالي خطوات إجرائية تحويل صيغة ما جيدة التركيب إلى شكل العطف النظامي.

لإيضاح الإجرائية سنستخدم الصيغة التالية: $\neg (P \supset Q) \vee (R \supset P)$

1. حذف إشارات الاقتضاء وذلك باستخدام الشكل المكافئ مع $\vee$:

$$\neg (\neg P \vee Q) \vee (\neg R \vee P)$$

2. تضييق مجال إشارات النفي $\neg(P \supset Q)$ تصبح $\neg P \vee Q$ باستخدام قانوني دومرغان، وبحذف إشارتي النفي المتتاليتين:

$$(P \wedge \neg Q) \vee (\neg R \vee P)$$

3. استخدام قوانين التجميع والتوزيع

$$\neg(P \wedge Q) = (\neg P) \vee (\neg Q)$$
$$\neg(P \vee Q) = (\neg P) \wedge (\neg Q)$$

للوصول إلى شكل العطف النظامي، فنحصل على:

$$(P \vee \neg R \vee P) \wedge (\neg Q \vee \neg R \vee P)$$

ثم على:

$$(P \vee \neg R) \wedge (\neg Q \vee \neg R \vee P)$$

نعبر عادةً عن عطف من العبارات بمجموعة من العبارات (العطف ضمني بين العبارات):

$$\{ (P \vee \neg R) , (\neg Q \vee \neg R \vee P) \}$$

الحل بالنقض

يتم الحل بالنقض لإثبات صحة صيغة **wff** من مجموعة من الصيغ Δ باتباع الخطوات التالية:

- تحويل كل صيغة، في Δ إلى عطف عبارات (شكل العطف النظامي).
- تحويل نفي الصيغة **w** المطلوب إثباتها إلى عطف عبارات.
- تجميع العبارات الناتجة عن الخطوتين السابقتين في مجموعة واحدة Γ.
- نعاود تطبيق الحل على عبارات المجموعة Γ ونضيف ناتج الحل في كل مرة إلى المجموعة ، حتى الوصول إلى حالة لا يبقى فيها عبارات قابلة للحل أو الوصول إلى المجموعة الخالية.

مثال

لتكن لدينا المعارف التالية:

(1) **BAT_OK** البطارية بحالة جيدة

(2) ¬**MOVES** الربوط لا يتحرك

(3) **BAT_OK** ∧ **LIFTABLE** ⊃ **MOVES** البطارية بحالة جيدة والربوط قادر على الحمل فإن الربوط سيتحرك.

يعطي تحويل العبارة الثالثة إلى الشكل النظامي:

(4) ¬**BAT_OK** ∨ ¬**LIFTABLE** ∨ **MOVES**

نريد أن نبرهن أن الغرض لا يمكن رفعه ¬**LIFTABLE**، يعطي نفي الصيغة المطلوب برهانها العبارة التالية:

(5) **LIFTABLE**

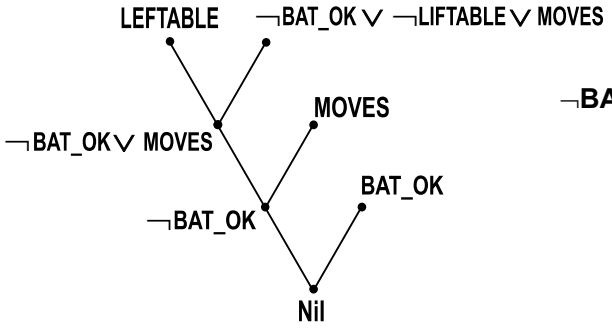
لنطبق الآن الحل للحصول على العبارات التالية:

(6) بحل (4) و(5) نحصل على ¬**BAT_OK** ∨ **MOVES**

(7) بحل (2) و(6) نحصل على ¬**BAT_OK**

(8) بحل (1) و(7) نحصل على **Nil**

نبين شجرة الحل في الشكل الجانبي.



استراتيجيات الحل بالنقض

على الرغم من سهولة الحل بالنقض، ولكن يجب اختيار إستراتيجية تبين ماذا علينا أن نحل أولاً.

هناك استراتيجيات ترتيب الحل:

- الحل عمقاً أولا،
- أم عرضاً أولاً.

واستراتيجيات تشذيب Refinement:

- ضرورة احتواء الحل على نفي ما مطلوب برهانه أو عبارة خلف لها،
- أو ضرورة احتواء الحل على عبارة من المجموعة الابتدائية (قبل إجراء أي حل)،
- أو ضرورة تحقيق الشرطين السابقين.

عبارات هورن

هي العبارات التي تحتوي حرفياً موجباً واحداً على الأكثر ولها ثلاثة أنواع:

- العبارة الوحيدة التي تحوي ذرة وحيدة وندعوها حقيقة **Fact**، مثل **P** و**Q**.
- الاقتضاء (و الذي ندعوه عادةً قاعدة **Rule**). تتألف مقدمته من عطف حرفيات موجبة ونتيجته من حرفي موجب وحيد، مثل **P∧Q⊃R**.

- مجموعة من الحرفيات السالبة والمكتوبة على شكل اقتضاء مقدمته تتألف من عطف حرفيات موجبة ونتيجته المجموعة الخالية. يمكن الحصول على هذا الشكل مثلاً عند نفي صيغة **wff** مطلوب إثباتها تتألف من عطف حرفيات موجبة. ندعو هذه الصيغة عادةً بالهدف **Goal**، مثل **$P \wedge Q \supset$.**

وبذلك نرى أن حساب الفرضيات يسمح بتمثيل نظام خبير يعتمد قواعد الإنتاج، إذ يمكن تمثيل الحقائق والقواعد، والأهداف التي نود برهنتها. وتكون المحاكمة باستخدام الحل والبرهان بالحل بالنقض.

3. الإسناديات predicates

تعتبر لغة حساب الفرضيات محدودة. إذ لا يمكننا مثلاً التعبير عن حقيقة أنه عندما نحرك الكتلة **B** مثلاً، فإن **B** هي الكتلة نفسها التي تؤكد الفرضيةُ **ON_B_C** أنها موضوعة على الكتلة **C**. ففي حساب الفرضيات تكون الذراتُ سلاسل محرفية ليس لها بنية داخلية. والفرضيتان **ON_A_B** و**ON_B_C** في عالم الكتل مثلاً، مختلفتان تماماً وليس بينهما شيء مشترك. ومع أننا سمينا هذه الذرات بأسماء ذات دلالة (لتساعدنا على تذكر المقصود بها)، فلا شيء يمنعنا من استخدام أسماء أخرى لها مثل **P124** و**Q23**.

من المفيد استخدام لغة تدل على الأغراض في العالم (مثل الكتل)، وعلى الفرضيات المتعلقة بالعالم . أي إننا نحتاج إلى لغة فيها تسميات للأغراض التي نضع لها الفرضيات، وللفرضيات الموضوعة. ففي عالم الكتل مثلاً، يمكن أن تكون لدينا فرضيات مثل **$\neg$CLEAR_C $\supset$ ON_B_C**، حيث تعني **CLEAR_C** عدم وجود أي شيء على الكتلة **C** (أي إن وجود الكتلة **B** فوق **C** تقتضي ألا يكون سطح **C** خالياً). يلزمنا للتعبير عن حقيقة مثل هذه لجميع كتل العالم، الكثير من الفرضيات. لذا من المناسب أن يكون بإمكاننا مجرد قول شيء مثل **$\neg$Clear (y) $\supset$On (x,y)**، حيث **x** و**y** متغيران يمكن أن يعبرا عن أية كتلة.

اللغة

هي لغة تضم ثوابت غرضية إضافة إلى ثوابت علاقاتية وتابعية، لدينا ما يلي:

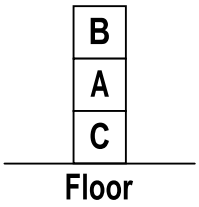
- مجموعة غير منتهية من الثوابت الغرضية **object constants**. سنستخدم لترميزها سلاسل محرفية، عادةً ما تكون لها دلالة تُساعدنا على تذكر مدلولها ، مثل : **Aa**, **13B**, **John**, **EiffelTower**.
- مجموعة غير منتهية من الثوابت الدالية **function constants** من جميع الأبعاد ، مثل :

distanceBetween(Damascus, Homs)
times (5, 4)

- مجموعة غير منتهية من الثوابت العلاقاتية **relation constants** من جميع الأبعاد (نسمي الثوابت العلاقاتية أحياناً الإسناديات)، من الأمثلة على الثوابت العلاقاتية : **Son(Radwan, Anas)**.
- مجموعة من المتغيرات، مثل:**times (4, x)**
- مجموعة الروابط ∨، ∧، ¬، ⊃ و∀ و∃ والمحددات [،]، **(،)** والفاصلة **،** .

يمكن استخدام حساب الإسناديات لتمثيل المعرفة، كما في المثال التالي:

- **Block (A)**
- **Block (B)**
- **Block (C)**
- **Floor (F1)**
- **On (B,A) , On (A,C)..**
- **Clear (B)**



حيث يبين الشكل التالي التقابل بين العالم وبين الإسناديات التي اتخذناها لتمثيله:

حساب الإسنادية	العالم
A	A
B	B
C	C
F1	Floor
On	On = {<B,A>,<A,C>,<C,Floor>}
Clear	Clear = {<B>}

قواعد الاستدلال

إن قواعد الاستدلال التي استخدمناها (مثل: مودس بوننس، وإدخال العطف وحذفه، وإدخال الفصل، وحذف النفي والحل) في حساب الفرضيات تبقى صحيحة ويضاف إليها قواعد أخرى نبينها فيما يلي:

- **الاستنساخ العام Universal Instantiation:** أي إنه من (∀x)P(x,f(x),B) نستطيع أن نستنتج P(A,f(A),B)، إذا تحققت إسنادية مهما تكن x، فهي تتحقق من مستنسخ A من x.
- **التعميم الوجـودي Existential Generalization:** أي إنه من (∀x)Q(A,g(A),x) نستنتج (∃y)(∀x)Q(y,g(y),x).

يمكن باستخدام دلالة المكميات المعطاة إثبات التكافؤات التالية، والمشابهة لقوانين دومورغان:

¬(∀ξ) ω (ξ) ≡ (∃ξ) ¬ω (ξ) إذا لم تكن صيغة محققة مهما يكن متحولها فهذا يعني أنه يوجد مثال لا تتحقق من أجله الصيغة.

¬(∃ξ) ω (ξ) ≡ (∀ξ) ¬ω (ξ) إذا لم يوجد مثال تتحقق الصيغة من أجله فهذا يعني أنها غير محققة مهما يكن متحولها.

(∀ξ) ω (ξ) ≡ (∀η) ω (η) يمكن تغيير المتحول الأصم للمكمي الشمولي ∀ دون أن تتأثر دلالة الصيغة.

التوحيد

إذا حوت عبارتان حرْفيْن متطابقين ومتتامين (أحدهما نفي الآخر)، فيمكن أن نحلهما بطريقة حساب

الفرضيات نفسها. وإذا كان لدينا حرفي $\lambda(\xi)$ في عبارة (حيث ξ متغير) وحرفي متمم $\neg\lambda(\tau)$ في عبارة أخرى (حيث τ حد لا يحوي المتغير ξ) فيمكن التعويض عن ξ بـ τ في العبارة الأولى، ثم القيام بحل الفرضيات على الحرفيات المتتامة للوصول لناتج حل **Resolvent** العبارتين.

ليكن لدينا مثلاً العبارتان **P(f(y),A)∨Q(B,C)** و**¬P(x,A)∨R(x,C)∨S(A,B)**. وبالتعويض عن **x** بـ **f(y)** في العبارة الثانية نحصل على **¬P(f(y),A)∨R(f(y),C)∨S(A,B)**. نلاحظ أن العنصرين في أول كل عبارة متتامان تماماً، ومن ثَم يمكن الحل على هذا الحرفي **P(f(y),A)** لنحصل على ناتج الحل **R(f(y),C)∨S(A,B)∨Q(B,C)**.

يجري حساب التعويضات المناسبة باستخدام إجرائية تدعى التوحيد **unification**. وهي هامة جداً للذكاء الصنعي.

يمكن لحدود تعبير أن تكون متغيرات أو ثوابت غرضية أو تعابير وظيفية. نحصل على منتَسخ التعويض **substitution instance** لتعبير ما، بالتعويض عن متغيرات التعبير بحدود. لذا فإن للتعبير **P[x,f(y),B]** منتسخات التعويض التالية:

P[z,f(w),B] **P[x,f(A),B]** **P[g(z),f(A),B]** **P[C,f(A),B]**

يدعى أول منتَسخ من اليسار بمتغير ألفبائي **alphabetic variant** لحرفي أصلي، لأننا لم نفعل سوى التعويض عن المتغيرات التي تظهر في التعبير **P[x,f(y),B]** بمتغيرات أخرى. يدعى آخر منتَسخ بالمنتسخ الأساسي **ground instance** إذ لا تحوي حدوده أي متغير.

يمكن تمثيل أي تعويض بمجموعة مرتبة من الثنائيات **s={**$\tau_1/\xi_1, \tau_2/\xi_2, ..., \tau_n/\xi_n$**}** حيث τ_i/ξ_i يعني أن كل ظهور للمتغير ξ_i سيعوض عنه بالحد τ_i على مدى التعويض. لاحظ أنه لا يمكن التعويض عن متغير بحد يحوي هذا المتغير. في التعبير السابق استعملنا التعويضات التالية للحصول على المنتسخات الأربعة:

التعبير	**P[x,f(y),B]**			
منتسخات التعويض	**P[C,f(A),B]**	**P[g(z),f(A),B]**	**P[x,f(A),B]**	**P[z,f(w),B]**
التعويضات المستخدمة	**s4 ={C/x,A/y}**	**s3 ={g(z)/x,A/y}**	**s2 ={A/y}**	**s1 ={z/x,w/y}**

يمكن تركيب تعويضين **s1** و**s2** وندون ذلك **s1s2**. للقيام بذلك نطبق أولاً **s2** على حدود **s1** ثم نضيف أي زوج من **s2** يحوي متغيرات غير موجودة في متغيرات **s1**. يمكن بيان أن تطبيق **s1** و**s2** بالترتيب على أي تعبير ω هو تطبيق **s1s2** نفسه على ω. أي **(**ω**s1)s2=**ω**(s1s2)**. كذلك يمكن بيان أن تركيب التعويضات تجميعي، أي **(s1s2)s3=s1(s2s3)**.

مثال

ليكن لدينا الصيغة ω التي هي **P(x,y)**، والتعويضان **s1={f(y)/x}** و**s2={A/y}** فيكون:

ω(s1s2)=[P(x,y)]{f(A)/x,A/y}=P(f(A),A) و **(ωs1)s2=[P(f(y),y)]{A/y}=P(f(A),A)**

لا تكون التعويضات عموماً تبديلية: أي لا يكون عموماً **s1s2=s2s1**. فترتيب تطبيق التعويضات يؤدي إلى اختلاف النتيجة.

مثال

باستخدام ω و**s1** و**s2** من المثال السابق

ω(s1s2)=P(f(A),A)
ω(s2s1)=[P(x,y)]{A/y,f(y)/x}=P(f(y) ,A)

عندما يطبق التعويض **s** على كل عنصر في مجموعة من التعابير **{ω_i}** ندون مجموعة منتسخات التعويض بـ **s{ω_i}**. نقول إن مجموعة **{ω_i}** من التعابير قابلة للتوحيد **unifiable** إذا وجد تعويض **s** بحيث يكون **ω_1s = ω_2s = ω_3s =...** ونقول في هذه الحالة إن **s** هو الموحد **unifier** لـ **{ω_i}** ذلك أنه يضم مجموعة الصيغ في أحادية **singleton**. على سبيل المثال ، **s={A/x,B/y}** يوحد المجموعة **{P[x,f(y),B],P[x,f(B),B]}** ليعطي **{P[A,f(B),B]}**. ومع أن **s={A/x,B/y}** هو موحد للمجموعة**{P[x,f(y),B],P[x,f(B),B]}** فإنه ليس الموحد الأبسط، إذ نلاحظ أننا لسنا مضطرين لإبدال (للتعويض عن) **x** بـ **A** لبلوغ التوحيد. نقول إن موحداً **g** لـ **{ω_i}** هو الموحد الأعم (أو الأبسط) **most general unifier (mgu)**، إذا كان أي موحد آخر **s** لـ **{ω_i}** يعطي **s{ω_i}** فحينئذ يوجد تعويض آخر **s'** بحيث **gs'{ω_i}=s{ω_i}**. فضلاً على ذلك، المنتسخ المشترك الناتج عن الموحد الأعم وحيد (ماعدا اختلافات أبجدية).

يوجد العديد من الخوارزميات المستخدمة لإيجاد الموحد الأعم لمجموعة منتهية من التعابير القابلة للتوحيد (وهي تعلن الإخفاق إذا كانت التعابير غير قابلة للتوحيد).

الخوارزمية **UNIFY** المستخدمة هنا، مأخوذة عن أحد المراجع. وهي تعمل على تعابير موضوعة في قائمة مهيكلة **list-structured**، وكل حرفي وكل حد فيها مكتوب كقائمة. فمثلاً الحرفي **(¬P(x,f(A,y))** يكتب بالشكل المهيكل **(¬Px(f A y))**، ويكون التعبير **¬P** هو التعبير الأول من المستوى الأعلى والتعبير **(f A x)** هو التعبير الثالث من المستوى الأعلى.

نستخدم في الخوارزمية **UNIFY** المفهوم الأساسي وهو مجموعات عدم التوافق **disagreement set**. نحصل على مجموعة عدم التوافق لمجموعة غير خالية ***W*** من التعابير بتحديد موقع أول رمز (بدءاً من اليسار) لا تكون عنده رموز كل التعابير في ***W*** متطابقة، ثم نستخرج من كل تعبير في ***W*** التعابير الجزئية التي تبدأ بالرمز الذي يحتل هذا الموقع. تكون مجموعةُ هذه التعابير الجزئية المتتالية مجموعةَ عدم التوافق لـ ***W***. على سبيل المثال، إذا كانت **{(¬Px(f z B) , (¬Px(f A y))}** هي مجموعة قائمتين

فإن مجموعة عدم التوافق لها هي **{A,z}**. يمكن جعل مجموعة عدم التوافق متوافقة باستخدام التعويض **A/z**.

إن الخوارزمية **UNIFY** إما توجد الموحد الأعم لمجموعة تعابير قابلة للتوحيد، وإما تعلن الإخفاق إذا كانت التعابير غير قابلة للتوحيد. ونتيجة الخوارزمية هي الموحد الأعم وهو زوج من تعابير القوائم المهيكلة.

مثال: التعويض **s={A/x,B/y}** يوحد المجموعة:**{P[A,f(y),B],P[x,f(B),B]}** ليعطي **{P[A,f(B),B]}**.

UNIFY (Γ)	حيث Γ مجموعة قوائم مهيكلة من التعابير
1 **$k \leftarrow 0$; $\Gamma_k \leftarrow \Gamma$; $\sigma_k \leftarrow \varepsilon$**	الخطوة الابتدائية. (ε هو التعويض الفارغ)
2 **if Γ_k is singleton, exit with σ_k, the mgu of Γ, otherwise continue**	إذا كانت Γ_k أحادية، أُخرُجْ من الإجراء مرجعاً σ_k وهي الموحد الأعم لِـ Γ، وإلا فتابع
3 **$D_k \leftarrow$ the disagreement set of Γ_k**	نضع في D_k مجموعة عدم التوافق لِـ Γ_k
4 **If there exits elements v_k, t_k in D_k such that v_k is a variable that does not occur in t_k continue. Otherwise exit with failure; Γ is not unifiable**	إذا وجد عنصران v_k وt_k في المجموعة D_k حيث v_k متغير لا يظهر في t_k تابع، وإلا فاخرُج وأعلِن الإخفاق، وتكون Γ غير قابلة للتوحيد
5 **$\sigma_{k+1} \leftarrow \sigma_k\{t_k/v_k\}$; $\Gamma_{k+1} \leftarrow \Gamma_k\{t_k/v_k\}$**	لاحظ أن $\Gamma_{k+1} = \Gamma_k\sigma_{k+1}$
6 **$k \leftarrow k+1$**	أضِف واحداً إلى **k**
7 **Go to 2**	اذهب إلى الخطوة **2**

الحل

لتطبيق الحل هنا، بهدف المحاكمة على المعرفة نحتاج أولاً إلى تحويل الصيغ الجيدة التركيب **wff** إلى عبارات ويتم ذلك باتباع الخطوات التالية:

- حذف الاقتضاء (عبارة مكافئة مع $\vee$، مثلاً نستعيض عن $x \Rightarrow y$ بـ $\neg x \vee y$)
- تقليص مدى النفي (تطبيق قوانين دومورغان)
- تقييس المتغيرات (لكل مكمي وجودي رمز خاص به)، مثلاً يمكن كتابة الصيغة **$(\forall x)[\neg P(x) \vee (\exists x)Q(x)]$** هكذا **$(\forall x)[\neg P(x) \vee (\exists y)Q(y)]$**.
- حذف المكميات الوجودية (باستخدام دوال سكوليم)، **مثال:** **$(\forall x)(\exists y)$ Height(x,y)** تصبح **$(\forall x)$ Height(x,h(x))**.
- التحويل إلى الشكل الأمامي **Prenex Form** (وضع جميع المكميات في المقدمة)
- وضع المصفوفة في شكل العطف النظامي
- حذف المكميات الشمولية
- حذف رمز العطف $\wedge$

➢ إعادة تسمية المتغيرات بحيث لا يظهر المتغير في أكثر من عبارة

بعد إجراء هذه التحويلات نستخدم الحل بالنقض كما في حالة حساب الفرضيات.

الحل

لنفترض أن الربوط يعرف أن جميع الطرود الموجودة في الغرفة **27** أصغر من أي طرد موجود في الغرفة **28**. أي:

$$\forall(x,y)\{package(x)\wedge package(y)\wedge inroom(x,27)\wedge inroom(y,28)\}\supset smaller(x,y)$$

تصبح

$$\neg P(x)\vee\neg P(y)\vee\neg I(x,27)\vee I(y,28)\vee S(x,y)$$

حيث: **P = package, I = inroom, S = smaller**

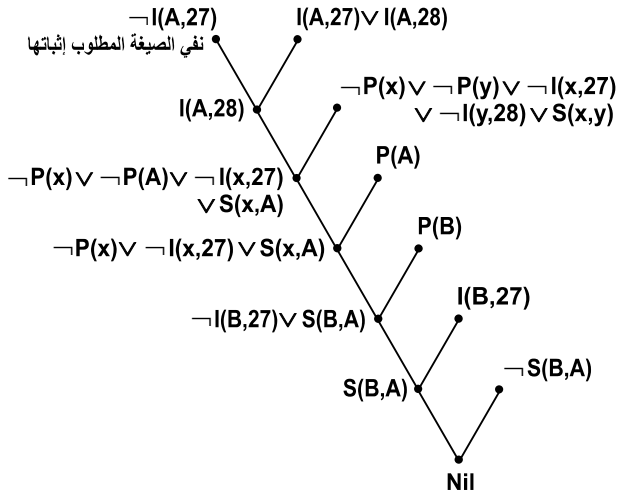
فإذا كان لدى الربوط الحقائق التالية:

P(A),P(B) و **I(A,28)** و **I(A,27)** و **I(B,27)** و **¬S(B,A)**

أي **A** و**B** أغراض و**A** موجود في الغرفة **27** أو الغرفة **28** والغرض **B** في الغرفة **27** و**B** ليس أصغر من **A**.

المطلوب: إثبات أن **A** في الغرفة **27** بطريقة الحل بالنقض.

تبين الشجرة طريقة الحل هذه.



الحل بالنقض

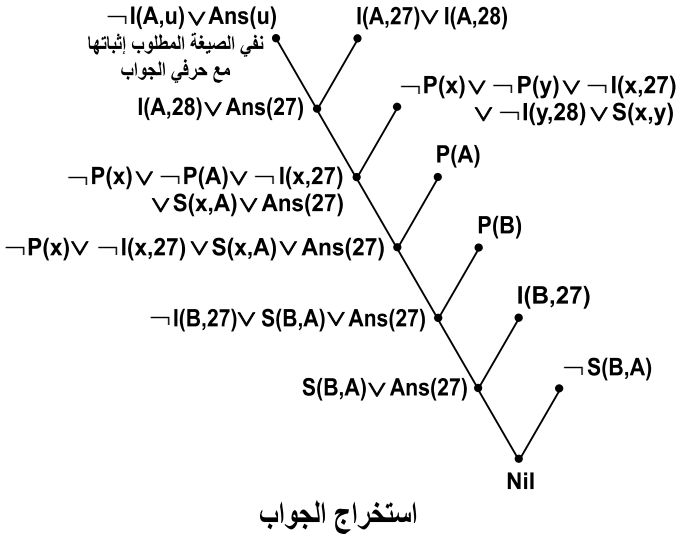
استخراج الجواب

قد نتطلب إيجاد الحل. أي لا نثبت الحل بنقض الفرض وإنما نبحث عنه. نريد إيجاد كل التعويضات التي أدت إلى الوصول إلى الحل (أي إعطاء تبرير لهذا الحل).

في المثال السابق نود أن نجيب على السؤال (في أي غرفة يوجد الطرد **A**). لذلك ، نضع **Ans (u)** ونريد أن نبرهن على: **(∃u) I(A,u)** أي إذا كان الجواب **u**، نود أن نوجد الغرفة **u** التي فيها الغرض **A**.

شجرة الحل هي المبينة.

نلاحظ أن الوصول إلى الحل، أعطى أن الغرفة **27** هي التي تتضمن الغرض **A**.



استخراج الجواب

4. تمارين

1. لدينا أزواج التعابير التالية، بين أقابلةٌ هي للتوحيد أم لا. ثم احسب الموحد الأعم لكل منها:
   1. **P(x,B,B) و P(A,y,z)**
   2. **P(g(f(v)),g(u)) و P(x,x)**
   3. **(x,f(x)) و P(y,y)**
   4. **P(y,y,B) و P(z,x,z)**
   5. **x=2+3 و x=3+3**
2. اشرح لماذا لا يكون **P(f(x,x),A)** قابلاً للتوحيد مع **P(f(y,f(y,A)),A)**.
3. حول الصيغ التالية إلى شكل العبارات **clause form**:
   1. **((∃-x)[P(x)]∨ (∃x)[Q(x)])⊃(∃x)[P(x)∨Q(x)]**
   2. **(∀x)[P(x)⊃(∀y)[(∀z)[Q(x,y)]⊃¬(∀z)[R(y,x)]]]**
   3. **(∀x)[P(x)]⊃(∃x)[(∀z)[Q(x,z)]∨(∀z)[R(x,y,z)]]**
   4. **(∀x)[P(x)⊃Q(x,y)]⊃((∃y)[P(y)]∧(∃z)[Q(y,z)])**
4. ليكن لدينا المقطع التالي:

“Tony, Mike, and John belong to the Alpine Club. Every member of the Alpine Club is either a skier or a mountain climber or both. No mountain climber likes rain, and all skiers like snow. Mike dislikes whatever Tony likes and likes whatever Tony dislikes. Tony likes rain and snow.”

"**Tony** و**Mike** و**John** أعضاء في نادي **Alpine**. كل عضو في هذا النادي يمارس رياضة التزلج على الثلج أو تسلق الجبال أو كلتا الرياضتين. لا يوجد متسلق جبال يحب المطر، وجميع المتزلجين يحبون الثلج. يكره **Mike** كل ما يحبه **Tony** ويحب كل ما يكرهه **Tony**. يحب **Tony** المطر والثلج."

مثل هذه الحقائق باستخدام حساب الإسناديات بحيث يمكن تمثيل السؤال التالي:

"Who is a member of the Alpine Club who is a mountain climber but not a skier?"

"من هو العضو في النادي الذي يمارس التسلق إلا أنه لا يتزلج؟"

كتعبير في الحساب الإسنادي. استخدم الحل بالنقض مع استخراج الجواب للإجابة عن هذا السؤال.

5. لنفترض أننا نعرف الحقائق التالية عن الفيلة الثلاثة **Sam** و**Clyde** و**Oscar**:

1. **Sam is pink.**
2. **Clyde is gray and likes Oscar.**
3. **Oscar is either pink or gray (but not both) and likes Sam.**

1. لون سام **Sam** وردي.
2. لون كلايد **Clyde** رمادي وهو يحب **Oscar**.
3. لون أوسكار **Oscar** إما وردي وإما رمادي (ولكن ليس كليهما)، وهو يحب **Sam**.

استخدم الحل بالنقض لإثبات أنه يوجد فيل رمادي يحب فيلاً ورديا. أي أثبت أنه:

$$(\exists x,y)[Gray(x) \wedge Pink(y) \wedge Likes(x,y)]$$

5. مراجع البحث

- كتاب "الذكاء الصنعي رؤية جديدة" من مطبوعات الجمعية العلمية السورية للمعلوماتية **2004**
- **Giarratano & Riley "Expert Systems: Principles and Programming", 3d Edition, 1998**

الفصل الثالث: طرق أخرى لتمثيل المعرفة والنظم الخبيرة

1. نظم قواعد الإنتاج

عرفنا سابقاً عبارات هورن التي تحوي حرفياً موجباً على الأكثر. ورأينا أن لدينا:

- قواعد (علاقات اقتضاء)
- حقائق
- أهداف نود تحقيقها (برهنتها)، بناء على القواعد والحقائق المتوفرة.

تسمح لغة **Prolog** ببرمجة مثل هذه النظم بسهولة.
يمكن أيضاً استخدام لغة **Clips** لتحقيق ذلك.

يوجد العديد من النظم الخبيرة التي تعتمد هذا الأسلوب:

لدينا قاعدة المعرفة التي تتضمن الحقائق (في ذاكرة العمل)، والقواعد في قاعدة المعرفة، ولدينا الأهداف (في ذاكرة العمل)، ولدينا محرك استدلال يسعى إلى إجراء محاكمة لمعرفة صلاحية الأهداف.

يمكن أن يعمل محرك الاستدلال بالسلسلة الأمامية **Forward chainig** انطلاقاً من الحقائق الموجودة ، أو بالسلسلة الخلفية **Backward Chaining** انطلاقاً من الهدف.

عبارات هورن

هي العبارات التي تحتوي حرفياً موجباً واحداً على الأكثر ولها ثلاثة أنواع:

- العبارة الوحيدة التي تحوي ذرة وحيدة وندعوها حقيقة **Fact**. (مثل **P** و**Q**)
- الاقتضاء (و الذي ندعوه عادةً قاعدة **Rule**). تتألف مقدمته من عطف حرفيات موجبة ونتيجته من حرفي موجب وحيد. (مثل $P \wedge Q \supset R$)
- مجموعة من الحرفيات السالبة والمكتوبة على شكل اقتضاء مقدمته تتألف من عطف حرفيات موجبة ونتيجته المجموعة الخالية. يمكن الحصول على هذا الشكل مثلاً عند نفي صيغة **wff** مطلوب إثباتها تتألف من عطف حرفيات موجبة. ندعو هذه الصيغة عادةً بالهدف **Goal**. (مثل $P \wedge Q \supset$)

السلسلة الأمامية

انطلاقاً من الحقائق الموجودة في ذاكرة العمل، يجب البحث عن القواعد التي يمكن تطبيقها واستنتاج حقائق جديدة.

مثال: إذا كانت لدينا القواعد التالية:

إذا كان **x** فيل فإن **x** من الثديات: $elephant(x) \supset mammel(x)$

إذا كان **x** من الثديات فإن **x** حيوان: **mammel(x)⊃animal(x)**

والحقيقة:

Clide هو فيل: **elephant(clide)**

بالسلسلة الأمامية نرى أنه يمكن تطبيق القاعدة الأولى وبتوحيد **x** مع **clide** نستنتج أن **mammel(clide)** أي إن **clide** من الثديات.

ثم بتطبيق السلسلة الأمامية ثانية وتطبيق القاعدة الثانية ، وتوحيد **x** مع **clide** نستنتج أن **animal(clide)** أي إن **clide** حيوان. وبذلك يضاف إلى ذاكرة العمل الحقيقتان : **mammel(clide)** و **animal(clide)**.

تُستخدم السلسلة الأمامية في تشخيص الأمراض، حيث إن تأكيد وجود مرض يضيف حقائق أعراض المرض إلى ذاكرة العمل.

السلسلة الخلفية

انطلاقاً من الهدف الذي نود تحقيقه، يجب البحث عن الفرضيات التي تدعم تحقيق هذا الهدف.

ففي المثال السابق إذا اعتبرنا القواعد نفسها موجودة وهي:

إذا كان **x** فيل فإن **x** من الثديات: **elephant(x)⊃mammel(x)**

إذا كان **x** من الثديات فإن **x** حيوان: **mammel(x)⊃animal(x)**

وهدفنا أن نبرهن أن **clide** حيوان أي: **animal(clide)**

نبحث عن قاعدة نتيجتها الوصول إلى **animal(clide)** فنجد أن القاعدة الثانية تؤدي إلى استنتاج هذا ، إذا تحقق لدينا **mammel(clide)** فيصبح الهدف الجديد الآن البحث عن فرضيات لبرهان الهدف الجديد **mammel(clide)**. نجد أنه بتطبيق القاعدة الأولى نصل إلى هذا الهدف إذا كان لدينا **elephant(clide)**. وبما أن هذه الحقيقة موجودة في ذاكرة العمل، فإن فرضية البدء صحيحة أي **animal(clide)** صحيحة.

مقارنة بين السلسلة الأمامية والسلسلة الخلفية

	السلسلة الأمامية	السلسلة الخلفية
الاستخدام	التحكم، التخطيط	التشخيص
مقاد بـ	المعطيات	الهدف
المحاكمة	من الأسفل إلى الأعلى، إيجاد النتائج التي تدعمها المعطيات	من الأعلى إلى الأسفل، إيجاد حقائق تدعم فرضيات
يشبه خوارزميات البحث	عرضاً أولاً	عمقاً أولاً

مشاكل قواعد الإنتاج وحلول

- غالباً لا يكون التمثيل البسيط فعالاً.
- لا يمكن التعبير عن بعض المعارف ببساطة باستخدام هذه القواعد.
- يصعب فهم وصيانة هذه القواعد حين تغدو ذات عدد ضخم جداً.

في البحث عن طرق تمثيل أخرى. نحتاج إلى تمثيل الأغراض المعقدة في العالم وخواصها وعلاقاتها بالأغراض الأخرى، فنبحث عن:

- تمثيل جيد مناسب لجميع المعارف في النطاق مهما تعقدت.
- إمكان استنتاج معارف أخرى من التمثيل.
- أن يحمل التمثيل ما يساعد على جعل الاستدلال فعالاً (التركيز باتجاه المعرفة الضرورية).
- تحصيل المعرفة على نحو فعال.

سنحتاج إلى معارف أكثر شمولاً لإنجاز نظم ذكاء صنعي، متعددة البراعات، من المستوى البشري . ستحتاج مثل هذه النظم إلى معارف عن العديد من المواضيع التي برهن على صعوبة نمذجتها صورياً. ولربما كان من المتناقضات أن علماء الذكاء الصنعي يجدون تمثيلَ المواضيع، السهلة الفهم على طفل في العاشرة من عمره، أشد مقاومةً لطرائق الذكاء الصنعي من تمثيل المواضيع التي أمضى الخبراء سنوات في تحصيلها. إذ يستطيع الفيزيائيون إعطاء وصف دقيق ومفَصل لظواهر فيزيائية باستخدام المعادلات الموجية ونظرية النسبية وبنى رياضية أخرى، أما الباحثون في الذكاء الصنعي فمازالوا يتجادلون حول أفضل طريقة لتمثيل حقائق بسيطة (إلا أنها جد مفيدة) مثل كون السائل يطابق شكل الفنجان الذي نضعه فيه، وأنه ينسكب إذا قلبنا الفنجان. ويبدو أن المميزات المغرقة في النظرية، التي ابتكرها الفيزيائيون والرياضيون أسهل صوغاً من الأفكار التي مكنت البشر برغم كل شيء، من أداء وظائفهم أداء جيداً تقريباً، حتى قبل ظهور أرسطو.

قد يكون أحد أسباب صعوبة تمثيل المعارف البديهية هو مجرد ضخامة هذه المعارف. يمكن تقسيم الكثير من المعارف الخبيرة بحيث تكفي المئات أو الآلاف من الحقائق لبناء نظم خبيرة مفيدة . كم سنحتاج من الحقائق لبناء نظام يصل إلى مستوى الذكاء البشري العام؟ لا أحد يعرف على وجه اليقين! يعتقد دوغ لينا **Doug Lenat** الذي يقود جهوداً لبناء قاعدة معرفة واسعة لمثل هذه الحقائق ، اسمها **CYC**، أننا سنحتاج إلى ما بين مليون وعشرة ملايين من هذه الحقائق.

تكمن مشاكل أخرى لصياغة مثل هذه المعارف في أنه ليس من السهل التعبير عن بعض المواضيع بجمل تصريحية. على سبيل المثال، ليس من السهل وصف الأشكال والأجسام الفيزيائية المعقدة باستعمال الجمل. فمثلاً، هل يمكن وصف وجه إنسان بكلمات، بحيث يمكن أن يتعرفه شخص آخر لم

يره قبلاً؟ كيف يمكن أن نستخدم الكلمات لوصف شجرة أو منظر جبلي أو غروب شمس مداري؟ وإذا كنا غير قادرين على وصف شيء باستخدام الإنكليزية أو أية لغة طبيعية أخرى، فهذا سبب كاف لنعتقد أننا لن نجد نمذجة لهذه المفاهيم بلغة المنطق أيضاً.

2. شبكات الدلالة

لا يختلف إلا القليلون حول قيمة "ربوط منزلي". هذه الآلة التي يمكن أن يعهد إليها بمهام تنظيف المنزل وغَسل الثياب وتحضير الوجبات وتقديمها، وإنجاز إجراءات الصيانة المنزلية (مثل تبديل مصابيح الإضاءة المحترقة)، وغَسل الأطباق، وغيرها. ولكن، فكر في كمية المعارف التي سيحتاج إليها هذا الربوط! سيحتاج إلى معارف كافية لتمكّنه من الإجابة عن أسئلة مثل:

أين تَضع مصابيح الإضاءة المحترقة؟
كيف تُمسك بصحن؟ أو بكأس؟
أين تضع جميع الأشياء التي تُخرجها من غسالة الصحون؟
كيف تُخبرنا بأن المكنسة الكهربائية بحاجة إلى تفريغ؟
كم من الوقت يمكن أن يبقى الخس، عادة، في الثلاجة؟

وغيرها وغيرها.

يمكن القول أيضاً أن المعارف البديهية هي التي تجعل النظام الخبير أكثر فائدة. ذلك أن النظم الخبيرة تنجز عملها جيداً في مجالات خبرتها فقط. إلا أن إضافة المعارف البديهية إليها يجعلها قادرةً ، على الأقل، على أن تدرك المطلوب حين يطلب المستخدم معلومات خارج تلك المجالات. إضافة إلى ذلك ، تمكّن هذه المعارف النظام من أن يتنبأ بدقة:

- متى تكون المعرفة لها صلة بالمهمة التي ينفذها الربوط؟
- ومتى لا تكون كذلك؟

يمكن، في كثير من الأحيان، ترتيب كيانات المعارف البديهية ومعارف مجالات الخبرة في بنى تراتبية تنظم عملية المحاكمة وتسهلها. ففي تمثيل المعارف البديهية في النظام **CYC**، على سبيل المثال، يشار إلى الكيان الأساسي جداً بالثابت الغرضي **Thing** (شيء). يوجد في **CYC** عدة أنواع من الأشياء **things**: أغراض في العالم، الأغراض الرياضية، الأحداث والإجرائيات وغيرها. تُرتب هذه الأشياء في تصنيف (أو تراتب) يرمز ضمنياً حقائق من الشكل:

" X is a P, all P's are Q's, All Q's are R's"

"**X** هي من النوع **P**، كل عناصر **P** هي عناصر من **Q**، و كل عناصر **Q** هي عناصر من **R**" وغيرها.

يمكن تمثيل تراميز التراتبات التصنيفية باستخدام الشبكات أو باستخدام بنى معطيات تُدعى الأطر **frames**. سنناقش أولاً التمثيل باستخدام الشبكات باستعمال معلومات عن الآلات المكتبية لتكون مثالاً توضيحياً.

لنفترض أننا نريد مثلاً تمثيل الحقائق التالية: **Snoopy** هي طابعة ليزرية، كل الطابعات الليزرية هي طابعات، كل الطابعات هي آلات، إضافةً إلى بعض المعلومات الأخرى المتعلقة بهذه المسألة. يمكن أن يكون لدينا عبارات الحساب الإسنادي التالية:

Laser_Printer(Snoopy)
(∀x) [Laser_Priner(x) ⊃ Printer(x)]
(∀x) [Priner(x) ⊃ Office_Machine(x)]

تُمثل كل من **Laser_Printer** و **Printer** و **Office_Machine** فئات **categories** في التراتبية.

- ينطوي أحد أنواع المحاكمة المهمة التي تستعمل المعرفة التصنيفية على تعدي الفئات . فمثلاً يمكننا، من الحقائق والقواعد السابقة، استنتاج ما يلي:

[Laser_Priner(x) ⊃ Office_Machine(x)] (∀x) و Office_Machine(Snoopy)

- يمكن أن يكون لجميع عناصر الفئات التصنيفية خواص معينة. فمثلاً مصدر الطاقة لجميع آلات المكتب هو مأخذ كهربائي في الجدار. يمكن التعبير عن هذه الخواص باستخدام دالّة وإسنادية المساواة:

(∀x)[Office_Machine(x) ⊃ [Energy_Source(x) = Wall_Outlet]]

- لاحظ أن عناصر الفئات الجزئية ترث نموذجياً الخواص المشتركة لفئاتها العليا:

(∀x)[Laser_Printer(x) ⊃ [Energy_Source(x) = Wall_Outlet]]

تُسهل هذه الأنواع العامة من الاستدلالات المتعلقة بالكيانات في التراتبية التصنيفية، باستعمال تمثيل بياني يسمى شبكة الدلالة **Semantic network**.

تعريف:

شبكات الدلالة هي بنى بيانية، تُستخدم لترميز المعارف التصنيفية عن الأغراض وخواصها.

سنُناقش فيما يلي نسخة مبسطة منها وذلك بهدف إظهار الأفكار الرئيسية.

يوجد نوعان من العقد:

- العقد المعنونة بثوابت علاقاتية وهي تُقابل فئات تصنيفية أو خواص.
- العقد المعنونة بثوابت غرضية تُقابل أغراضاً في المجال.

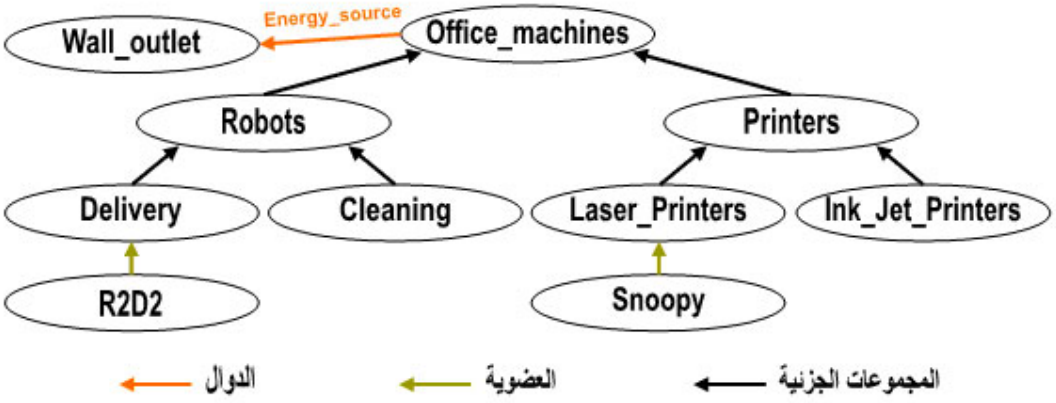
كما يوجد ثلاثة أنواع من الأسهم تربط هذه العقد:

- أسهم المجموعات الجزئية (والتي تُدعى عادةً **isa links**، أي روابط من النوع **isa**)

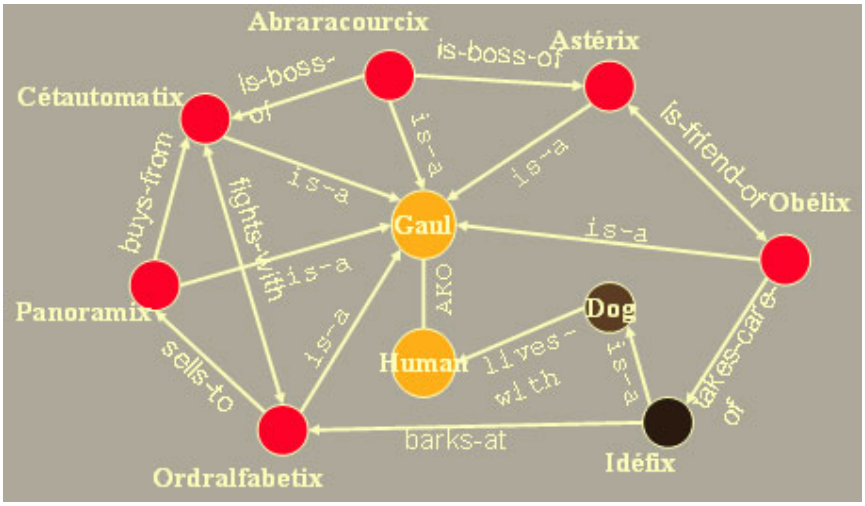
- أسهم العضوية (الانتماء إلى مجموعة) (والتي تُدعى عادةً روابط منتسخات أو عينات).
- أسهم الدوال.

مثال على شبكة دلالة.

مثال (1)



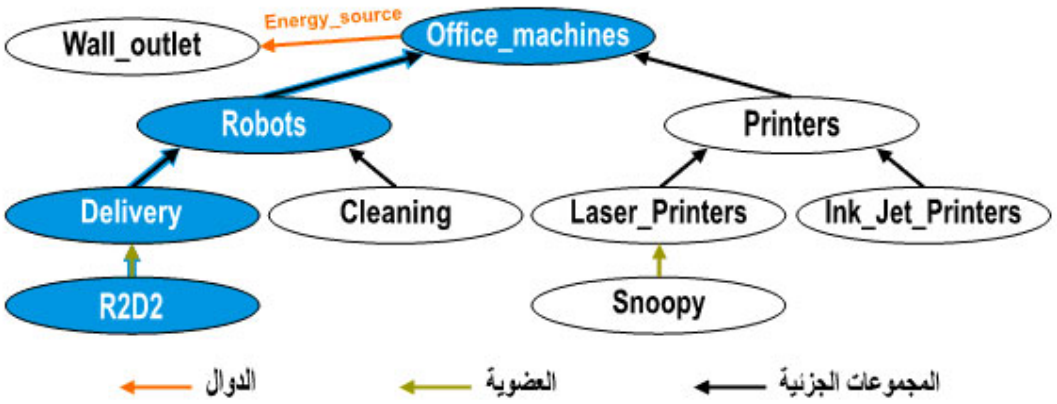
مثال (2)



إن المحاكمة المتعلقة بالخواص أو الانتماء إلى مجموعة أسهل وأشد فاعلية باستخدام الشبكات الدلالية من المحاكمة باستخدام الحل غير الموجه على الصيغ الجيدة التركيب الموافقة.

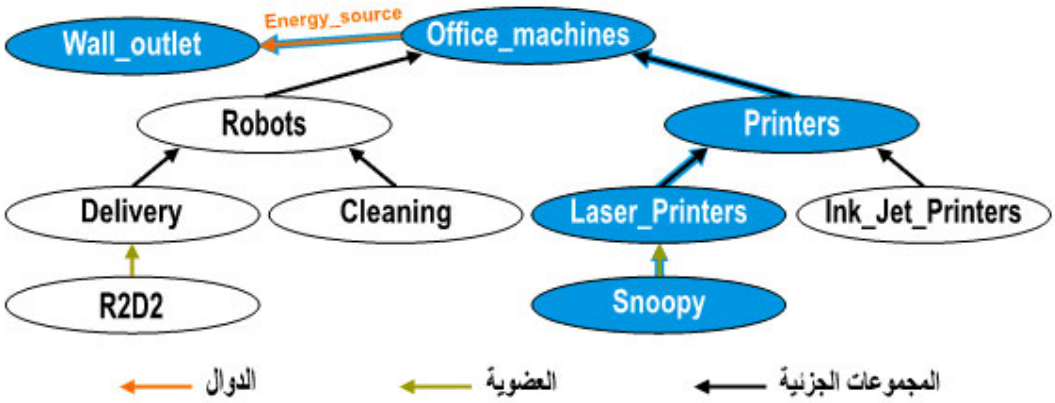
لكي نصل إلى قرار بشأن غرضٍ ممثلٍ بعقدة **A**: أهو عضو من مجموعة ممثلة بعقدة **B**، يجب اتباع الأسهم صعوداً من **A** لنرى: هل سنقابل العقدة **B** (سنصل إلى **B**). **مثال**

يمكننا بسهولة، أن نقرر أن الآلة **R2D2** هي آلة مكتبية **Office_Machine** على سبيل المثال.



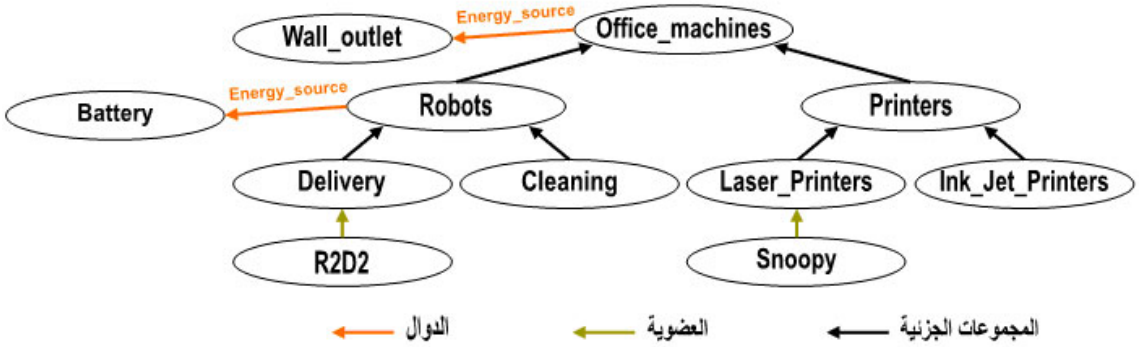
ولتعيين قيمة إحدى الخواص لغرض ممثل بعقدة **A** نتبع الأسهم المنطلقة من **A** صعوداً، بحثاً عن عقدة لها هذه الخصيصة. **مثال**

لتعيين مصدر الطاقة لِ **Snoopy** نتبع الأسهم المنطلقة من **Snoopy** صعوداً إلى العقدة **Office_Machines** ونجد أن الجواب الممثل هناك هو **Wall_outlet** أي مأخذ في الجدار.



إننا نقوم في غالب الأحيان باستدلالات مغتَفَلة **by default**؛ وهي استدلالات تحاول حذف المعرفة التي تؤدي إلى نقيض ما نرغب في افتراضه صحيحاً. ولكن إذا وصلت معارف جديدة مناقضة، فعلينا أن نقوم بسحب الاستدلال المغتَفَل. لقد اقتُرِح العديد من النظم والآليات المنطقية للإمساك بهذه الظاهرة ، أبسطها إلغاء التوريث.

لنفترض، في المثال السابق، أننا نريد القول بأن مصدر الطاقة المغتفَل للآلات المكتبية ، هو مأخذ كهربائي في الجدار. ولكن يوجد استثناء لهذه القاعدة، وهو أن مصدر طاقة الربوط هو البطارية. يمكن التعبير عن هذه المعرفة بإضافة سهم وظيفي آخر إلى الشبكة الدلالية كما هو مبين في الشكل.



قد يبدو أن آلية وراثة الخواص ستقود إلى تناقض في الشبكة الجديدة. مع أن الوراثة تسمح لنا باستنتاج (وهذا صحيح) أن مصدر طاقة الطابعات هو مأخذ كهربائي جداري، فإنها تُناقض أيضاً الحقيقةَ التصريحية الممثلة بحقيقة أن مصدر طاقة الربوط هو البطارية (بافتراض أن البطارية والمأخذ الجداري مختلفان). إننا محميون من الوقوع في هذا التناقض بحسب الطريقة التي تُستَعمل فيها الشبكة. فللمعلومات عن أكثر الأصناف خصوصية (مرتبة حسب أقواس المجموعات الجزئية أو أقواس المنتسخات) أفضليةٌ على الأصناف التي هي أقل خصوصية. لذلك، فإننا في الحالة التي نحاول فيها الاستفسار عن مصدر طاقة **R2D2** نبحث أولاً عن موضع العقدة المعنونة **R2D2** في الشبكة، ثم ننظر: هل للعقدة **R2D2** سهم دلالي معنون بمصدر الطاقة؟ ففي حالة الإيجاب، نقبل الجواب المعطى بالعقدة الموجودة في رأس هذه القوس كعقدة نهائية. وإلا، فإننا سنتبع الأسهم بصعود تراتبي إلى أن نصل إلى أول عقدة لها سهم دلالي معنون بمصدر الطاقة، ثم نستعمل العقدة على رأس هذا السهم للجواب عن هذا الاستفسار.

إن المعلومات المرافقة للعقد العليا في التراتب التصنيفي (الهرمية) هي عادةً معلومات مغتَفَلة عامة ، مرشحة للإلغاء من قبل معلومات أكثر خصوصية مرافقة لعقد أدنى في التراتبية.

فكما يظهر الشكل، معلومات مصدر الطاقة **Energy_source** للعقدة العليا **Office_machines** والمتمثلة بالمأخذ الجداري **Wall_outlet** هي معلومات مغتَفَلة عامة، تُلغى من قبل معلومات مصدر

الطاقة **Energy_source** الأكثر خصوصية للعقدة الأدنى **Robots** والمتمثلة بالبطارية **Battery**.

مواضيع تتعلق بالتنفيذ العملي

تمثيل شبكات الدلالة بسيط وفعال

- جداول تسرد الأغراض وخواصها
- جداول أو قوائم **lists** بالعلاقات

يمكن التحول منه إلى طرائق تمثيل أخرى

- في حساب الإسناديات:
  - العقد ثوابت أو متغيرات
  - الوصلات: إسناديات
- في حساب الفرضيات:
  - العقد والعلاقات هي فرضيات ترتبط بروابط منطقية مناسبة

قيمة الواصفة	الواصفة	الغرض
محارب	المهنة	**Astérix**
كبير جداً	الحجم	**Obélix**
صغير	الحجم	**Idéfix**
لا نهائية	الحكمة	**Panoramix**

يمكن إدراج المعارف كثلاثيات (غرض - واصفة - قيمة) OAV

- يمكن استخدام هذه الثلاثيات لتمثيل شبكات الدلالة
- تقود بسرعة إلى حجوم هائلة من المعلومات

مشاكل شبكات الدلالة

التعبيرية

- لا توجد بنى داخلية للعقد، فهي مجرد أغراض فقط.
- لا يوجد طريقة سهلة لتعريف التجريبيات (الكسبيات) **heuristics**.
- يمكن توسيعها لتعبر عن معارف أعقد وأعمق، ولكن هذا يؤدي إلى حجوم هائلة.
- ملائمة فقط للعلاقات الثنائية، وليس الثلاثية أو المتعددة.

الفعالية

- نصل بسرعة إلى عدد هائل من العقد والعلاقات.
- يؤدي البحث فيها إلى توسعات انفجارية خاصة إذا كانت نتيجة البحث سلبية.

استخدامها

- لا يوجد معايير لأنواع الوصلات، باستثناء وصلات التعريف **is-a** وربط الجزء بالكل **part-of** وربط الصفوف بصفوف آباء **a-kind-of** أو اختصاراً **AKO** ولا لتسميات العقد.
- تعتمد المعرفة المستنتجة على خاصية الوراثة، ولكن أحياناً توجد استثناءات في الصف الابن ، مثال: يتمتع صف الطيور بخاصة الطيران، صف الدجاج منه لا يطير

مثال: تحويل إلى شبكات دلالة

Human(Marc) ⟹	is-a(Marc, Human)
Score(Wahda, Itihad (75 73)) ⟹	is-a(G23, game) local_team(G23, Wahdah) visitor_team(G23, Itihad) score(G23, 75_73)

ملاحظة: شبكات الدلالة مفيدة في وصف دلالة اللغات كما في المثال:

"Ahmad gave the book to Mona"

is_a(Action, give);
Tense(Action, past);
Subject(Action, Ahmad)
Obj1(Action, Mona)
Obj2(Action, book).

3. ترابط المفاهيم

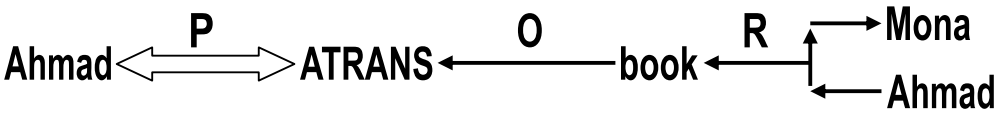
ترابط المفاهيم: Conceptual Dependencies

هو طريقة للتعبير عن دلالة جمل باللغة الطبيعية بحيث:

- نسهل استنتاج الاستدلال من الجمل.
- يكون التمثيل مستقلاً عن اللغة التي قيلت فيها الجملة (عربية أم انكليزية...).

طُورت هذه الطريقة في السبعينيات بواسطة **Schank** ونفذت في العديد من البرمجيات لفهم اللغات الطبيعية.

مثال: Ahmad gave the book to Mona (أحمد أعطى الكتاب إلى منى)



في هذا المثال:

- الأسهم تشير إلى اتجاه الترابط.
- السهم المزدوج يشير إلى علاقة باتجاهين بين الفاعل والفعل.
- **P**: تعني الزمن الماضي للفعل.
- **ATRANS**: أحد الأساسيات **Primitives** ويشير إلى نقل ملكية.
- **O**: غرض **object**.
- **R**: المستقبل **receiver**.

في ورشة العمل بعنوان **Arabic Natural Language Processing** التي أقيمت في المغرب في نيسان **April 2007** على هامش مؤتمر **ictis'07** (انظر الموقع: **www.ictis07.org**) جرى التصريح عن

Universal Networking Language (UNL) تعتمد المفاهيم لتمثيل اللغة العربية، وأنه جرى الاتفاق بين ممثلين عن **15** لغة للتمثيل بهذه اللغة. المشروع مدعوم من هيئة الأمم المتحدة.

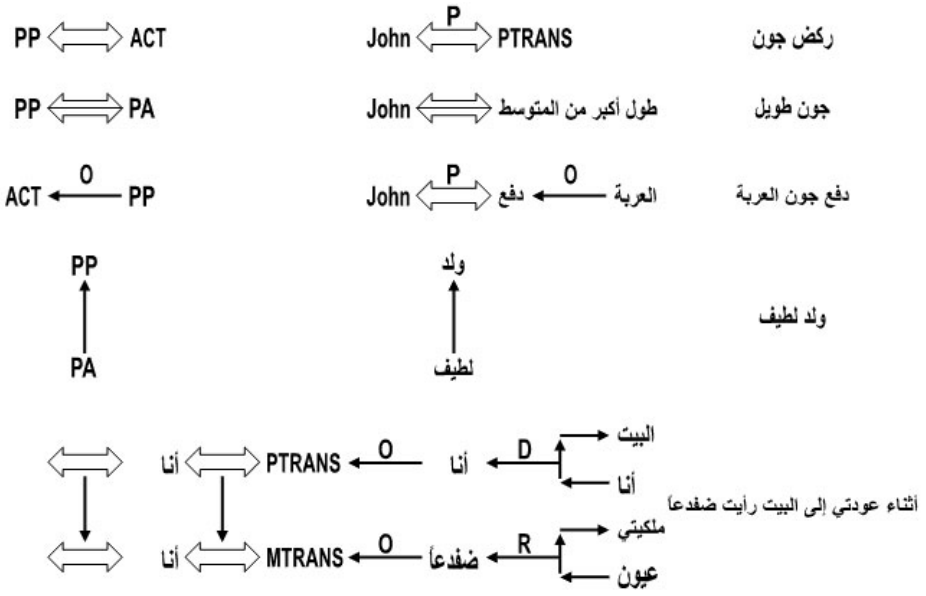
ثمة عدد من الأساسيات، يشير كل منها إلى مجموعة من الأفعال، وقد اعتمدت مجموعة الأساسيات التالية:

- **ATRANS**: **Abstract Transfer** أي نقل مجرد، مثل: أعطى **give**، أخذ **take**، سـرق **steal**، نهب، أهدى،...
- **PTRANS**: **Physical Transfer** أي نقل موضع فيزيائي لغرض، مثل: ذهب **go**، أتى، سافر ، قفز،...
- **PROPEL**: تطبيق قوة مادية على غرض، مثل: دفع **push**، كسر، حطم، رفع،...
- **MOVE**: تحريك جزء من الجسم من قبل صاحبه، مثل: ركل **kick**، لوح، غمز، ضم،...
- **GRASP**: التقاط غرض من فاعل، مثل: رمى **throw**، التقط، أمسكن...
- **INGEST**: هضم شيء، مثل: أكل **eat**، شرب، تناول دواء،...
- **EXPEL**: لفظ شيء، مثل: بكى **cry**، صرخ، نادى، أطلق،...
- **MTRANS**: نقل معلومات فكرية، مثل: قال **say**، روى، شرح، نبه، أدب، احتج، أثنى،...
- **MBUILD**: استنتاج معلومات من أخرى، مثل: قرر **decide**، محص، درس،...
- **SPEAK**: إنتاج أصوات، مثل: قال، غنى ,**sing**، قال، صرخ، عزف،...
- **ATTEND**: التركيز بالحواس، مثل أصغى **listen**، راقب، أمعن، حضر،...

هذه الأساسيات تحدد ترابطات بين كيانات على أربعة أنواع هي:

- **ACT**: **Actions**، الأفعال، وهي تندرج تحت الأساسيات السابقة (جرى، أخذ، أعطى، بكى،...)
- **PP**: **Picture Producers**، وتدل على منتج الصورة أو الفاعل (شخص، حيوان، نبات..)
- **AA**: **Action Assistant**، مساعد على تغيير الفعل، قد يكون هو المفعول أو غيره (قوة،..)
- **PA**: **Picture Assistant**، مساعد على تغيير الصورة، وهي الصفات مثل (طول، لطف،..)

أمثلة إضافية:



المحاكمة على المعرفة:

- تقسيم الأفعال إلى أساسيات يخفض من عدد قواعد الاستدلال (أعطى، أخذ، سرق ، وهب ...) تُعامل معاملة واحدة **ATRANS**.
- الكثير من الاستدلالات متضمن في التمثيل: يتضمن التمثيل أكثر مما تحمله كلمات العبارة.

مفاهيم الزمن (توضع على الأسهم):

P: الماضي **past**
f: المستقبل **future**
t: تحول أو انتقال **transition**
ts: التحول البدئي **start transition**
tf: تحول منتهٍ **finished transition**
K: مستمر **Continuous**
?: استفهامي
nil: الحاضر
Delta أو **D**: دائم **eternal**
/: نفي
c: شرطي **conditional**

تطبيقات هذا التمثيل:

- فهم اللغة الطبيعية للتطبيقات التفاعلية.
- المساعدة على فهم اللغة لأغراض الترجمة من لغة إلى أخرى.

صعوبة:

ليس هناك اتفاق على كل الأساسيات، أو على الأساسيات الواجب إضافتها لتحسين التمثيل ، أو على درجة تعقيد التمثيل..

لتمثيل المعارف الأكثر تعقيداً يجب أن:

- نوجد علاقة مباشرة بين الفعل أو الإدراك ونتيجته
- نمثل المعارف الضمنية العامة للناس

هذا التمثيل مرتبط بمفهوم **record** في البرمجة الحاسوبية وتجريد المفاهيم ضمن ترابط المفاهيم جزء من ذلك. مع أننا نحتاج إلى استدلالات خاصة بالعلاقات التي تربط المفاهيم.

من الأمثلة على تمثيل معارف أعقد:

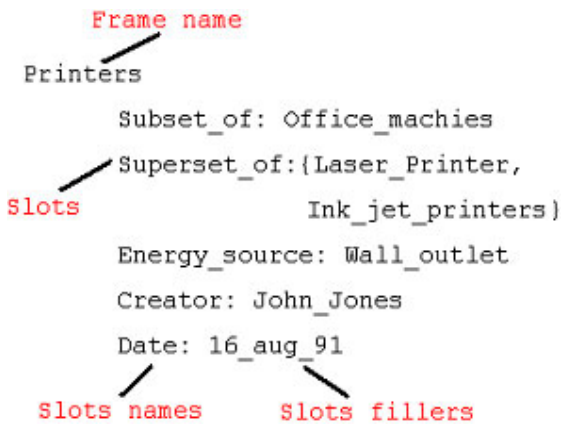
- الأطر (ملء شقوب **slots' filling**، وطرائق للملء)
- السيناريو (تسلسل مرتب زمنياً من الأطر)

4. الأطر Frames

هي نوع من شبكات الدلالة المعقدة، التي تتضمن بنية داخلية متطورة (معرفة مترفعة **meta-knowledge**)، صاحب الفكرة في إنشائها هو **Marvin Minsky** الذي قال في عام **1975**:

"حين نواجه موقفاً جديداً (أو نقوم بتغيير أساسي في رؤيتنا للمسألة) فإننا نختار من ذاكرتنا بنية تسمى الإطار، إنها إطار عمل يمكن تذكره، نسعى إلى ملاءمته ليناسب الحقيقة بتغيير تفاصيله إذا استدعى الأمر."

"When one encounters a new situation (or makes a substantial change in one's view of a problem) one selects from memory a structure called a 'frame'. This is a remembered framework to be adapted to fit reality by changing details as necessary."



للإطار اسم ومجموعة من الأزواج واصفة-قيمة. يقابل اسم الإطار عقدةً في شبكة دلالية ، أما الواصفات فتقابل أسماء الأسهم المرافقة لهذه العقدة، وتقابل القيم العقد في الطرف الثاني من الأسهم. نسمي عادة ، الأزواج واصفة - قيمة بالشقوب **slots**. ونسمي الواصفات أسماء الشقوب **slot names**، ونسمي القيم مالئات الشقوب **slot fillers**. ويبين الشكل مثالاً على إطار.

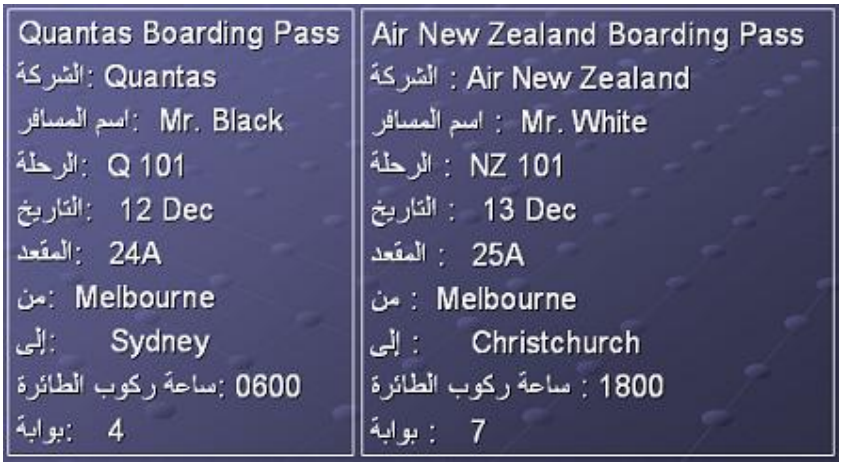
بالإضافة لذلك، يمكن تعريف قيم مغتفلة للشقوب **default values**، نعتمدها ما لم يأتنا ما يناقضها ، ويمكن أن نعرف حدوداً دنيا وعظمى للقيم بحسب نموذجنا الذهني، كما يمكن استدعاء إجرائيات برمجية عند تغيير قيمة الشقب أو عند الحاجة.

يمكننا أيضاً تمثيل بعض المعارف المترفعة **meta knowledge** في هذا التمثيل؛ على سبيل المثال ، تاريخُ إنشاء الإطار واسم من قام بإنشائه، وهي معارف عن الإطار، لا عن الطابعات.

أمثلة أخرى على الأطر:

هنا إطار لبطاقة أخذ مقعد في الطائرة، يتضمن الإطار واصفات (شقوب يجب ملؤها، يمكن استدعاء خوارزميات للملء، أو الإشارة إلى أطر أخرى):

- اسم شركة النقل،
- اسم الشخص،
- رقم الرحلة،
- تاريخ الرحلة،
- المقعد،
- مدينة الانطلاق،
- مدينة الوصول،
- ساعة الإقلاع،
- ورقم البوابة.



أهمية الأطر

في النظم الخبيرة التي تعتمد القواعد، المعرفة مبعثرة في قاعدة المعرفة، وعلينا البحث عن المعرفة غير البارزة، وهذا يبطئ البحث. أما باستعمال الأطر، فإننا نستخدم معرفتنا النموذجية عن الأغراض والحالات وتكون المعرفة على شكل أغراض لها واصفات نحدد قيمها بطرق مختلفة.

يمكن أن يمثل الإطار غرضاً أو مجموعة من الأغراض المتشابهة:

- الإطار الصف **Class-Frame**: مجموعة أغراض لها الواصفات نفسها. **Class of computers**
- الإطار المنتسخ (المثال) **Instance-Frame**: يشير إلى منتسخ معين من الصف. **My_computer**
- كل إطار يعلم إلى أي صف ينتمي ولكل إطار خواص ضمنية.

في المثال السابق: الإطار الصف هو المبين جانباً والأمثلة المعطاة هي منتسخات.

العلاقات بين الصفوف:

- يرتبط الصف الجزئي بالصف الأعم بعلاقة **is-a**.
- يمكن للإطار المنتسخ أن يعيد كتابة بعض القيم الخاصة به (مثال: النعامة من صف الطيور لكنها لا تطير).
- يبقى المنتسخ عضواً من صفه حتى وإن هتك بعض القيم النموذجية في الصف، وهذا يسمى التخصيص.

- الأطر من المستويات الأعلى تمثل مفاهيم مولدة **generic** أما من المستوى الأدنى فتمثل مفاهيم مقيدة.

العلاقات بين الأغراض:

- التعميم **Generalization**: **is-a, a-kind-of** (مثال: السيارة هي وسيلة نقل فهي من صف أعم).
- التجميع **Aggregation**: **part-of, whole-part** (مثال: المحرك جزء من السيارة).
- الربط **Association**: يصف علاقات دلالة بين مختلف الصفوف (مثال: يمتلك السيد **X** سيارة، ويمتلك السيد **X** كلب حراسة).

المحاكمة في الأطر:

- **الوراثة:** الإطار المنتسخ يرث **inherent** عن الإطار الأب (الصف)، ويمكن أن يرث من أكثر من أب (وراثة متعددة). لهذا الأمر سيئاته لأنه لا يميز بين الخواص الهامة والخواص العابرة، كما أن اتخاذ القرار صعب في حالات الوراثة المتعددة.
- **الطرائق methods والعفاريت demons!**: الطريقة إجرائية ترتبط بواصفة إطار وتنفذ حين الطلب. يمكن أن تحدد قيمة واصفة أو تنفذ حين تتغير قيمة واصفة. الـ **demon** يرتبط بواصفة عبارات القواعد **IF-THEN** وينفذ حين تتغير واصفات هذه العبارات. ثمة نوعان من الطرائق:
  - حين الحاجة **when-needed**
  - حين التغيير **when-changed**

الاستدلال في النظم المعتمدة على الأطر:

- يبحث محرك الاستدلال عن الأهداف **goals**.
- تلعب القواعد أدواراً غير كبيرة، معظم المعرفة ضمن الأطر.
- تضاف الطرائق والـ **demons** لإضافة أفعال إلى الأطر، ويمكن تحقيق الهدف بتنفيذ أي منها.
- إذا كان الاستدلال مقوداً بالمعطيات والأحداث فهو **bottom-up**.
- إذا كان الاستدلال مقوداً بالتوقعات فهو **top-down**.
- يبحث محرك الاستدلال عن القواعد (المتعلقة بالإطار **meta knowledge**) التي يكون الهدف نتيجتها ويبحث مدى صلاحيتها قاعدة تلو الأخرى **Backward Chaining**.
- الحصول على القيم من الطرائق "حين الحاجة **when needed**" إذا كان هذا مذكوراً في القواعد.
- البحث عن ترتيب طرق الحصول على قيم الواصفات (على سبيل المثال: ابحث عن القيمة من القواعد فإن فشلت نفذ طريقة "حين الحاجة"..)

مقارنة بين نظم القواعد ونظم الأطر:

في النظم الخبيرة التي تعتمد القواعد

- تمثل القواعد المعارف اللازمة لحل مشكلة.
- تلتقط كل قاعدة جزءاً من تجريبيات (كسبيات) المسألة.
- تضيف كل قاعدة معرفة جديدة إلى النظام فتجعله أذكى.
- يمكن تعديل قاعدة القواعد: بإضافة أو تغيير أو حذف قواعد.

في النظم الخبيرة التي تعتمد الأطر

- يجب أولاً تحديد البنية الهرمية للمعرفة.
- تحديد الصفوف والواصفات.
- بناء العلاقات التراتبية الهرمية بين الأطر.
- يسمح هذا البنيان بإضافة أفعال إلى الأطر عبر الطرائق **methods** والـ **demons**.

خطوات بناء نظام خبير يعتمد على الأطر:

1. تحديد المسألة ومدى مجال النظام.
2. تحديد الصفوف وواصفاتها.
3. تحديد المنتسخات.
4. تصميم الواجهة.
5. تعريف الطرائق والـ **demons** (حين الحاجة **when-needed** وحين التغيير **when changed**).
6. تعريف القواعد.
7. تقويم النظام وتوسيعه.

مثال على خطوات بناء نظام خبير يعتمد على الأطر: الشراء الذكي Buy Smart

- الخطوة الأولى: تحديد المسألة ومدى مجال النظام ابدأ بجمع معلومات عن الممتلكات (البيوت) المعروضة للبيع في المنطقة.

حدد تفاصيل مهمة بارزة:

- الموضع
- عدد غرف النوم والحمامات
- السعر
- وصف بسيط
- صورة جيدة للبيت

اسرد الأسئلة التي يمكن أن تتعرض لها:

- أعلى مبلغ يمكن رصده لشراء البيت

- نوع البيت الذي تفضله (شقة، بيت عربي،..)
- المنطقة التي تود العيش فيها
- عدد غرف النوم التي ترغب فيها
- عدد الحمامات التي ترغب فيها

➢ الخطوة الثانية: تحديد الصفوف والواصفات.

حدد الصفوف أو الأطر الصفوف:

- حدد مفهوم البيت **Property** ووصف عام لأهم مقوماته.
- ميز واصفات كل بيت: موضعه وسعره ونوعه وعدد غرف النوم، وصف، صورة،...
- تفاصيل للوصول إلى البيت (عنوان، رقم هاتف،...)

CLASS: *Property*
[Str] *Area*:
[Str] *Suburb*:
[N] *Price*:
[Str] *Type*:
[N] *Bedrooms*:
[N] *Bathrooms*:
[Str] *Construction*:
[Str] *Phone*:
[Str] *Pictfile*:
[Str] *Textfile*:
[N] *Instance Number*:

➢ الخطوة الثالثة: تعريف المنتسخات **Instances**.

أوجد المنتسخات باستخدام قاعدة معطيات أو وسط آخر للحصول على المعطيات:

INSTANCE: **Property 1**	
Class:	*Property*
[Str] *Area*:	دمشق
[Str] *Suburb*:	برزة
[N] *Price*:	1640000
[Str] *Type*:	شقة
[N] *Bedrooms*:	3
[N] *Bathrooms*:	1
[Str] *Construction*:	بناء حديث
[Str] *Phone*:	(03) 6226 4212
[Str] *Pictfile*:	*house01.bmp*
[Str] *Textfile*:	*house01.txt*
[N] *Instance Number*:	1

INSTANCE: **Property 2**	
Class:	*Property*
[Str] *Area*:	دمشق
[Str] *Suburb*:	مهاجرين
[N] *Price*:	1500000
[Str] *Type*:	شقة
[N] *Bedrooms*:	3
[N] *Bathrooms*:	1
[Str] *Construction*:	قديم
[Str] *Phone*:	(03) 6226 1416
[Str] *Pictfile*:	*house02.bmp*
[Str] *Textfile*:	*house02.txt*
[N] *Instance Number*:	2

➢ الخطوة الرابعة: تصميم الواجهات **Display**.

يجب أن تعرض واجهة عنوان التطبيق معلومات عامة في بداية كل استخدام:

- وصف عام للمسألة،
- صور بيانية موضحة،
- مع بعض معلومات حفظ حقوق الملكية.

- الخطوة الخامسة: تعريف الطرائق **methods** والـ **demons**: **When needed & When changed**.
- الخطوة السادسة: تعريف القواعد.

طرائق لجعل تطبيقنا حياً:

- باستخدام **when needed & when changed**.
- باستخدام قواعد مواءمة النماذج **pattern matching**.

حين يعتمد نظامنا على الأطر نفضل تعريف الطرائق والـ **demons** على قواعد مواءمة النماذج.

- الخطوة السابعة: تقويم النظام وتوسيعه.
  - أحد أهم القرارات التي يجب البت بها: هل نستخدم قواعد أم طرائق؟
  - غالباً يكون القرار تبعاً لذوق المصمم.
  - المثال السابق يستخدم الطرائق لأنها فعالة وسهلة لتمثيل الإجرائيات.

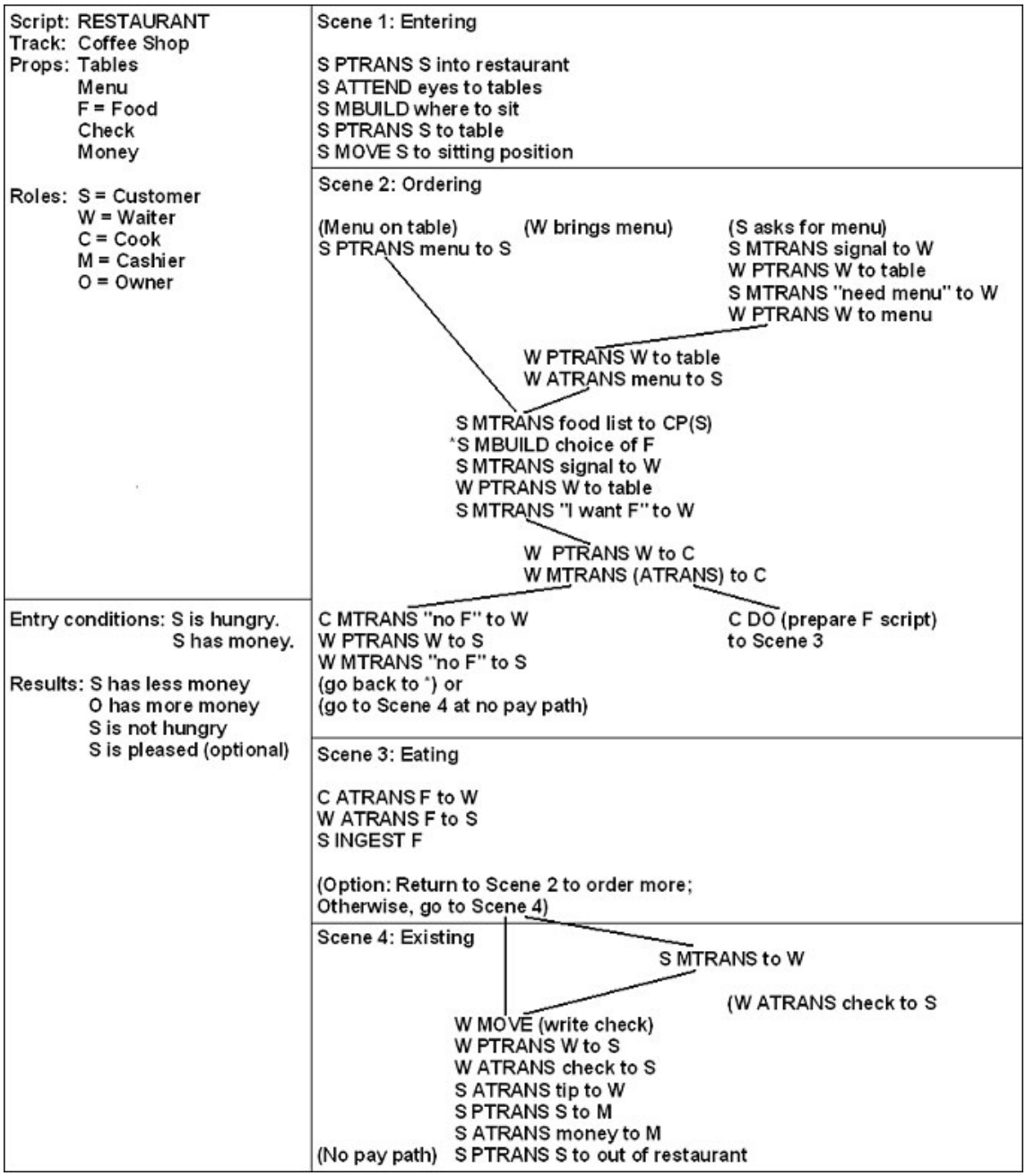
5. السيناريو

المشهد **script** أو **scenario**، بنية تصف تسلسلاً من الأحداث في سياق معين **context**. لدينا شقوب **slots** ولكل شقب قيمة، وقيمة مفترضة **by-default** إذا لم نتمكن من الحصول على القيمة . ولأن المشهد يخص سياقاً معيناً، فلدينا معلومات خاصة أكثر دقة حول البنية.

- في المثال التالي (عن **Schank, 1977**) مشهد أو سيناريو للذهاب إلى مطعم:
  - عنوان المشهد **Script**: هو المطعم
  - المكان: مقهى

- الأغراض: طاولات، لائحة الطعام، طعام **F**، شيك، نقود
- لاعبو الأدوار: الزبون **S**، النادل **W**، الطباخ **C**، المحاسب **M**، مالك المطعم **O**.
- شروط دخول المشهد: **S** جائع، و **S** لديه نقود
- نتائج المشهد: **S** نقوده أقل، **O** ازدادت نقوده، **S** لم يعد جائعاً، **S** مسرور (خياري)

أما تسلسل المشاهد فيبينها **الشكل التالي**.



المثال برمجياً

Generic RESTAURANT Frame

Specialization-of: Business-Establishment
Types:
  range: (Cafeteria, Fast-Food, Seat-Yourself, Wait-To-Be-Seated)
  default: Seat-Yourself
  if-needed: IF plastic-orange-counter THEN Fast-Food,
  IF stack-of-trays THEN Cafeteria,
  IF wait-for-waitress-sign or reservations-made THEN Wait-To-Be-Seated,
  OTHERWISE Seat-Yourself.
Location:
  range: an ADDRESS
  if-needed: (Look at the MENU)
Name:
  if-needed: (Look at the MENU)
Food-Style:
  range: (Burgers, Chinese, American, Seafood, French)
  default: American
  if-added: (Update Alternatives of Restaurant)
Times-of-Operation:
  range: a Time-of-Day
  default: open evenings except Mondays
Payment-Form:
  range: (Cash, CreditCard, Check, Washing-Dishes-Script)
Event-Sequence:
  default: Eat-at-Restaurant Script
Alternatives:
  range: all restaurants with same Foodstyle
  if-needed: (Find all Restaurants with the same Foodstyle) [Rogers 1999]

برمجة المشهد (السيناريو)

EAT-AT-RESTAURANT Script

Props: (Restaurant, Money, Food, Menu, Tables, Chairs)
Roles: (Hungry-Persons, Wait-Persons, Chef-Persons)
Point-of-View: Hungry-Persons
Time-of-Occurrence: (Times-of-Operation of Restaurant)
Place-of-Occurrence: (Location of Restaurant)
Event-Sequence:
  first: Enter-Restaurant Script
  then: if (Wait-To-Be-Seated-Sign or Reservations)
  then Get-Maitre-d's-Attention Script
  then: Please-Be-Seated Script
  then: Order-Food-Script
  then: Eat-Food-Script unless (Long-Wait) when Exit-Restaurant-Angry Script
  then: if (Food-Quality was better than Palatable)
  then Compliments-To-The-Chef Script
  then: Pay-For-It-Script
  finally: Leave-Restaurant Script

[Rogers 1999]

فائدة المشاهد

- إذا لم يكن المشهد أساسياً يمكن أن يكون لدينا مؤشراً إليه دون تفعيله.مثال:
(مرت سوزان أمام مطعمها المفضل في طريقها إلى المتحف. لقد أعجبت بلوحات بيكاسو)
- حين يكون المشهد أساسياً، نستطيع استنتاج معارف جديدة دون ذكرها صراحة لأن المشهد قد عرف بنية تسلسل المعارف والحوادث، المثال التالي يجيب على سؤال "هل أكل رامي البارحة؟" بـ "نعم":(ذهب رامي إلى المطعم، وطلب شواء، وعند الدفع اكتشف أنه لم يبق لديه كثير من المال، أسرع إلى المنزل لأن المطر بدأ بالهطول)
- هو طريقة لبناء تفسيرات وحيدة متماسكة ، بدءاً من مجموعة مشاهدات . فالمشاهد أو السيناريوهات هي سلسلة علاقة سببية للأحداث، المثال التالي يجيب على سؤال "لماذا أعطت النادلة أحمد لائحة الطعام؟" بـ "لأنه طلبها منها، أو لكي يتمكن من طلب الطعام دون أن يطلبها":(ذهب أحمد إلى الغداء، جلس إلى الطاولة ونادى النادلة. أعطته النادلة لائحة الطعام ، وطلب هامبرغر)
- تسمح المشاهد بتركيز الانتباه على أحداث غير اعتيادية، مثال:(ذهب سامي إلى مطعم، اقتيد إلى طاولة، طلب شواء، بقي ينتظر مدة طويلة، غضب وغادر) هنا انقطع السيناريو بعد طلب الطعام ولا يمكننا استنتاج شيء من الأحداث التي تتلو طلب الطعام من المشهد، لحدوث شيء غير اعتيادي وهو الانتظار الطويل، حيث لا نستطيع التنبؤ بما يمكن أن يحدث بعده.

6. تمارين

1. "ذهب سامي إلى المطار، لم يكن في عجلة من أمره لأنه يتوقع تأخر الطائرة . تناول بعض الشطائر والعصير، وكم كانت مفاجأته كبيرة حين علم بأن الطائرة كانت على وقتها."
   1. مثل الحقائق السابقة مستخدماً خمساً على الأقل من المفاهيم:

ATRANS, PTRANS, MTRANS, DIGEST, MOVE, GRASP, EXPEL, ATTEND, SPEAK

   2. هل كان سامي مسافراً؟ أم ذهب إلى المطار لهدف آخر؟. أجب على ذلك من خلال سيناريو **script** تضعه لدخول المطار تبين فيه:
      - مكان المشهد،
      - أغراضه،
      - اللاعبين،
      - شروط دخول المشهد،
      - نتائج المشهد،
      - تسلسل أحداث يقوم بها المسافر،

- و تسلسل أحداث يقوم بها من يذهب لاستقبال أو وداع شخص آخر.

2. "ذهبت الأسرة لتناول البوظة، لم يرد أحد الأطفال تناولها فرماها. تناول الآخرون البوظة وشعروا بالبرد فأسرعوا إلى السيارة وحدث بعضهم بعضاً: ما كان يجب تناولها الآن."
   1. مثل الحقائق السابقة مستخدماً أربعاً على الأقل من المفاهيم:
      ATRANS, PTRANS, MTRANS, INGEST, MOVE, GRASP, EXPEL, ATTEND, SPEAK
   2. هل كان الجو صيفاً أم شتاء؟ أجب على ذلك من خلال سيناريو **script** تضعه لتناول البوظة في مكان عام تبين فيه فقط:
      - شروط دخول المشهد،
      - اللاعبون،
      - نتائج المشهد،
      - وتسلسل أحداث يقوم بها اللاعبون (في الصيف وفي الشتاء).
3. نود بناء قاعدة معرفة لوجبات مطاعم متعددة، بغية تحديد الوجبة المناسبة بحسب متطلبات غذائية معينة. تتضمن المعرفة عن الوجبات:
   - المقبلات وما تتضمنه من خضار أو بقول أو صلصات،
   - الطبق الرئيسي هل هو حيواني أم نباتي،
   - طبق الحلوى (حلويات، أو فواكه أو كريمات)
   - كمية الحريرات (منخفض وسط مرتفع)،

   مثّل هذه المعارف باستخدام الأطر. وأعط مثالاً **instance** لوجبة تعرفها.

7. مراجع البحث

- محاضرات نظم قواعد المعرفة: كلية المعلوماتية **2006 – 2007**، د. أميمة الدكاك
- كتاب الذكاء الصنعية رؤية جديدة
- محاضرات ورشة العمل **ANLP** انظر الموقع **www.ictis07.org**
- **What is a Knowledge Representation**

1. فضاء الحالات

تحتاج آلة تفاعلية ذات مهارة واسعة إلى كمية هائلة من الذاكرة (يمكن أن يتعذر بلوغها) لكي تستطيع أداء مهام معقدة في بيئة معقدة. إضافة إلى ذلك، يجب على مصمم مثل هذه الآلة التفاعلية أن يتمتع برؤية ثاقبة لكي يحدد ردود الفعل المناسبة لجميع الحالات المحتملة. يقودنا ذلك إلى مراعاة الزمن بدلاً من الحجم، والتكيف عوضاً عن التصميم الصريح. يمكن أن نضيف إلى وظيفة الفعل بعضاً من الحسابات التي على مصمم الآلة التفاعلية إجراؤها. بالطبع، تتطلب هذه الحسابات زمناً، ولكنها تقلّص من احتياجات الذاكرة وتخفّف الأعباء على المصمم.

لنأخذ المثال التالي:

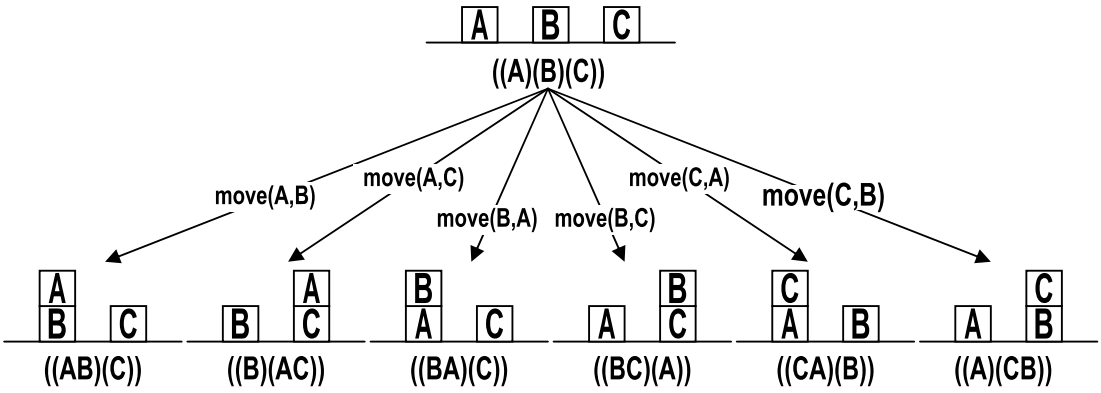
ليكن في عالم الفضاء-الشبكي ثلاثة مكعبات **A** و **B** و **C**، تكون جميعها في البداية على الأرض. ولنفترض أن مهمة الربوط أن يكدسها بعضها فوق بعض، بحيث يكون المكعب **A** فوق **B** والمكعب **B** فوق **C** والمكعب **C** على الأرض.

إن الأعمال الواجب أداؤها لتحقيق هذه المهمة بديهية بالنسبة لنا، في حين أنها ليست كذلك بالنسبة للربوط. لنفترض أن الربوط يستطيع نمذجة نتيجة كل فعل من أفعاله في بيئته. ويمكن أن يقوم بذلك بوساطة زوج من نماذج العالم:

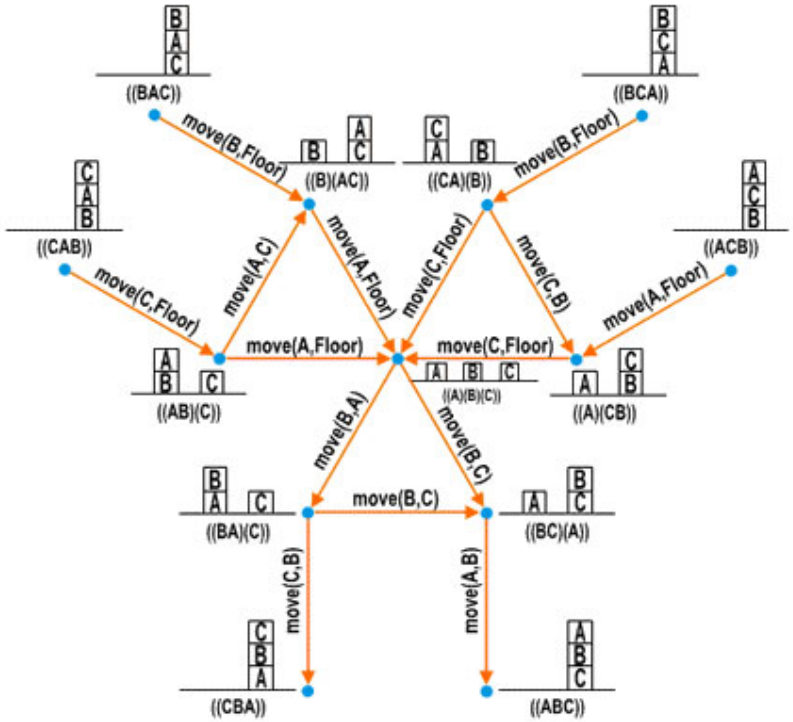
- يمثل النموذج الأول حالة العالم قبل تنفيذ الفعل،
- في حين يمثل النموذج الآخر حالة العالم بعد تنفيذ الفعل.

لتحقيق غاية هذا المثال، نفترض أن الربوط يستطيع تحريك أي مكعب، **x**، شريطة عدم وجود أي من المكعبات فوقه، إلى مكان آخر، **y**، إما على الأرض أو فوق مكعب لا يوجد فوقه أي مكعب آخر . تتغير البيئة لدى القيام بهذا التحريك **move(x, y)**، وهذا الفعل يسمى مؤثراً.

يبين الشكل التالي نتائج جميع الأفعال التي يمكن القيام بها في لحظة معينة.



في المحاكاة، يمكن أن ينتج عن النظر إلى الأمام خطوة واحدة فقط، توقعات مفيدة، غير أن استطلاع أكثر من خطوة، ربما على كامل الطريق حتى إتمام المهمة، يمكن أن يكتشف اختصارات ويسمح بتجنب المسارات العمياء. إن أنسب بنية لمتابعة نتائج البدائل العديدة من تسلسل الأفعال هي بنية البيان الموجه، الذي يبينه الشكل.



تعريف البيان:

يتألف البيان **Graph** من مجموعة من العقد **nodes** (ليست منتهية بالضرورة). تتصل بعض الأزواج من العقد فيما بينها بأقواس **arcs**، وهذه الأقواس موجهة من أحد عنصري الزوج إلى الآخر. نسمي مثل هذا البيان "**البيان الموجه directed graph**".

ولكي نحقق أهدافنا، نعنون العُقد بنماذج حالات العوالم، ونعنون الأقواس بأسماء الأفعال.

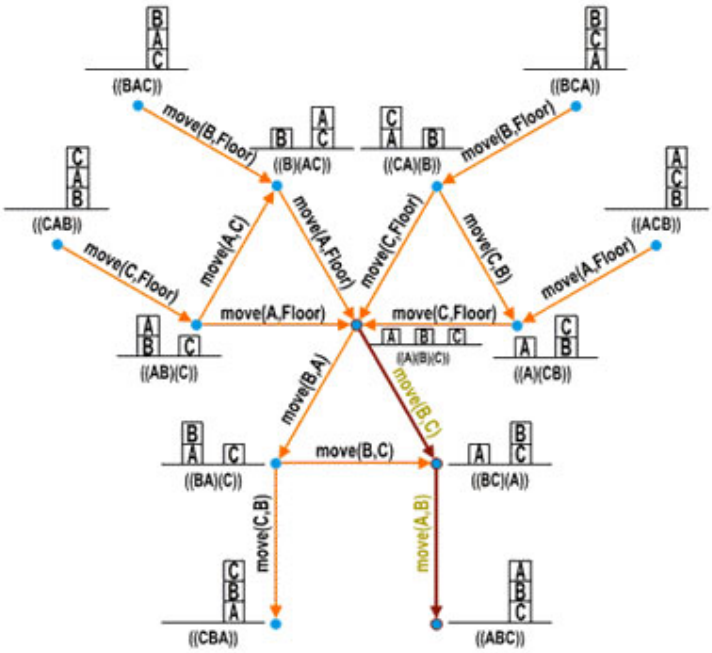
إذا كان القوس موجهاً من العقدة n_i إلى العقدة n_j، فإننا نسمي العقدة n_j **خَلَف successor (أو وليدة)** العقدة n_i، ونسمي العقدة n_i **سلفَ (أباً) parent** العقدة n_j. لا يمكن لعقدة، في البيان، أن تحوي سوى عدد منته من الخلَف.

يمكن أن تكون كل عقدة في زوج من العقد خَلَفاً للأخرى؛ في هذه الحالة، نستعيض عن زوج الأقواس **بوصلة**. يسمى البيان الذي لا يحوي سوى وُصلات **بياناً غير موجه**. تحمِل الأقواس في مخططات البيان اللاحقة أسهماً، في حين لا تحمل الوصلات ذلك.

1. فضاء الحالات

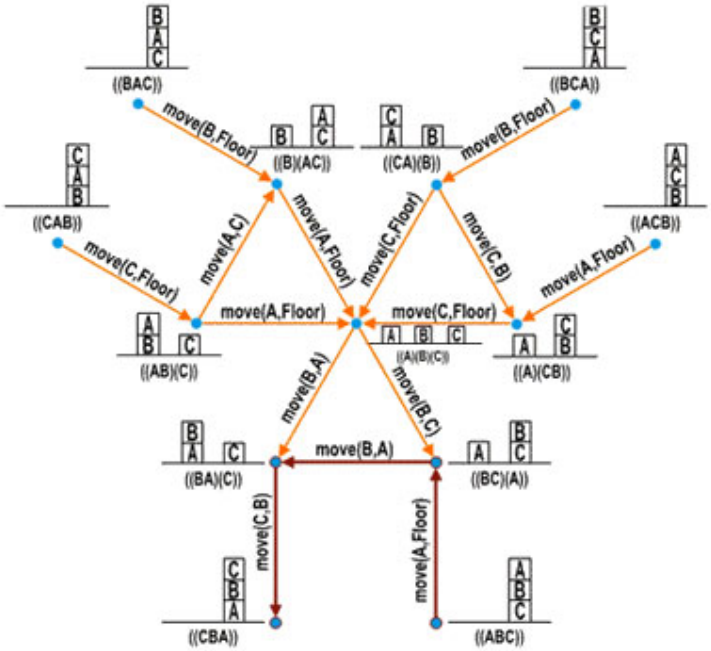
إذا كان عدد حالات العوالم المختلفة صغيراً إلى حد كاف، فعندها يمكن تخزين بيان كامل يمثل جميع الأفعال والحالات الممكنة. فمثلاً، يبين البيان الموضح في الشكل السابق جميع الحالات والحركات الخاصة بالتعامل مع ثلاثة مكعبات. نسمي مثل هذا البيان لنماذج العوالم والأفعال "**بيان فضاء الحالات**". لاحظ أن كل فعل من الأفعال هو عكوس، علماً أننا عنونّا فقط أحد زوجَي الحركات المتقابلة للتنقل بين التشكيلات، وذلك لجعل الصورة أقل فوضى. وانطلاقاً من البيان، من السهل ملاحظة أنه إذا كانت الحالة الابتدائية كالتالي: **((A)(B)(C))**، وإذا كانت المهمة هي الوصول إلى الحالة الموصوفة بـ**((ABC))**، فإن على الربوط أن ينفذ تسلسل الأفعال التالي:

{move(B,C),move(A,B)}



إن من مزايا تمثيل العوالم المحتملة ببنية البيان، أنه يمكن أن نأخذ أي عقدة في البيان لتمثيل حالة هدف : من الممكن أن يكون الهدف موصفاً من مصدر خارجي كالإنسان. ولإيجاد مجموعة أفعالٍ تُو صِلُ إلى هدف محدد، لا يحتاج الربوط إلا إلى إيجاد مسار في البيان من العقدة التي تمثل حالة العالم الابتدائية (عقدة البداية) إلى العقدة التي تمثل حالة الهدف المحدد (عقدة الهدف). ومن ثم يمكن قراءة الأفعال التي تحقق هذا الهدف على اللصيقات على الأقواس في هذا المسار. فمثلاً، إذا كانت مهمة الربوط الذي يكدس المكعبات أن يكدس المكعب **C** على **B** على **A** على الأرض، وكانت الحالة الابتدائية كالآتي: المكعب **A** على **B** على **C** على الأرض، فإن تسلسل الأفعال سيكون:

{move(A,Floor),move(B,A),move(C,B)}



يجب أن نكون قادرين على تمثيل جميع حالات العوالم الممكنة بوساطة عُقَد في بيان، وعلينا معرفة النماذج الدقيقة لكيفية تنفيذ الأفعال بين زوج من حالات العوالم. يجب على نظام إدراك الربوط، في مثالنا، أن يتعرف تعرفاً صحيحاً عقدةَ البداية. كما لا يمكن أن يوجد ربوطات أو عمليات ديناميكية أخرى يمكنها تغيير العالم. فإذا تحققت جميع هذه المسلمات، وإذا كان الزمن المخصص لهذه المهمة يكفي للبحث عن حالة الهدف، فإنه من الممكن التخطيط لكامل تسلسل الأفعال ومن ثم تنفيذه (تتابعياً، كما هو).

2. البحث في فضاء الحالات

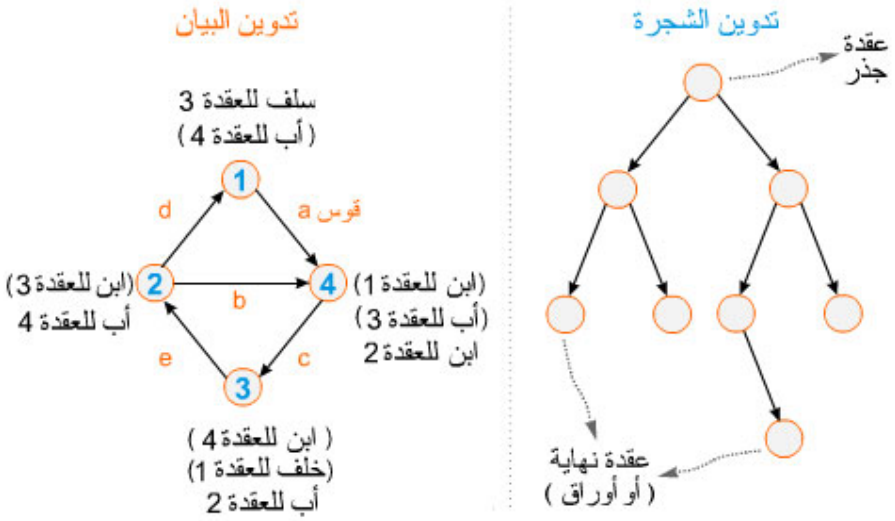
تتطلب طرائق البحث في البيانات الكاملة انتشار "مَعالِم" على عقد البيان. نبدأ بعنونة عقدة البداية بالـصفر **0**، ثم ننشر تدريجياً أعداداً أكبر على شكل أمواج على طول الأقواس إلى أن يصادف عدد ما، عقدة الهدف. ثم نرسم المسار رجوعاً من الهدف إلى البداية وفق سلسلة تنازلية من الأعداد. وتكون الأفعال على طول هذا المسار، من البداية إلى الهدف، هي الأفعال التي يجب إجراؤها لتحقيق الهدف . تحتاج هذه الطريقة إلى **O(n)** خطوة، حيث **n** هو عدد العقد في البيان.

عنونة العقد للبحث عن الهدف
(البحث الشامل)

نسمي عملية وضع علامة على خلْفِ عقدة ما "**التوسع** expansion". يضع التوسع علامات على جميع الجوار غير المعلمة للعقدة المعلمة. إن السؤال المهم المطروح هو: ما هي العقدة المعلمة (لكن غير الموسعة) التالية الواجب توسيعها؟ بطريقة البحث عرضاً_أولاً، العقدة التالية غير الموسعة الواجب توسيعها هي تلك المعنونة بعدد لا يتجاوز جميع عناوين العقد غير الموسعة الأخرى. بمعنى آخر، نوسع جميع العقد المعنونة بالعدد **i** قبل توسيع تلك المعنونة بالعدد **j** حيث **i < j**. وثمة طرائق بحث أخرى تطبق خيارات مختلفة لتوسيع العقد؛ سوف نشرحها بالتفصيل لاحقاً.

عموماً، فإن جميع استراتيجيات البحث عن حل تعمل على توليد الحالات الواجب سبرها، ثم اختيار الحالة المرشحة للمتابعة في كل لحظة، حتى الوصول إلى الهدف (الذي قد لا يكون وحيداً).

تعتبر **الشجرة الموجهة** (ذات الجذر) حالة خاصة من البيان الموجه، لكل عقدة فيه (عدا واحدة) أب وحيد فقط. وتسمى العقدة الوحيدة التي ليس لها أب **عقدة الجذر root**. وتسمى العقد في الشجرة التي ليس لها خلف **عقد النهاية** أو **عقد الأوراق**. ونقول إن **عمق depth** عقدة الجذر يساوي **الصفر**. نعرف عمق أي عقدة أخرى في الشجرة بأنه عمق الأب مضاف إليه **1**. وتعتبر **الشجرة (ذات الجذر) غير الموجهة** (مع وصلات عوضاً عن الأقواس) بياناً غير موجه فيه مسار واحد وواحد فقط عبر الوصلات بين أي زوج من العقد.



غالباً ما يكون من المناسب وضع تكاليف موجبة على الأقواس، وذلك لتمثيل تكلفة تنفيذ الفعل المقابل. ويكون البحث عن أقصر الطرق بين عقدتين أو أقلها تكلفة.

سوف ندرس صفين واسعين من إجرائيات البحث:

- في الصف الأول، لا يوجد سبب خاص بالمسألة لتفضيل جزء من فضاء البحث على آخر، وذلك فيما يخص إيجاد مسار إلى الهدف. توصف مثل هذه الإجرائيات بالعمياء **uninformed**.
- في الصف الآخر، توجد معلومات خاصة بالمسألة للمساعدة على تركيز البحث. توصف مثل هذه الإجرائيات بالتجريبية (الكسبية) heuristic.

تأتي كلمة التجريبية (الكسبية)**heuristic** من الكلمة اليونانية **heuriskein**، وتعني " يكتشف ". وتأتي الجملة المقتبسة **Eureka!** ("وجدتها!") من المصدر نفسه.

سوف نبدأ الآن بشرح الإجرائيات العمياء ونترك الإجرائيات التجريبية إلى فقرة قادمة.

إجرائيات البحث العمياء Uninformed

نجزئ العقد المولدة إلى مجموعتين:

- **Open**: تضم عقد بيان البحث التي تنتظر المعالجة. وتحوي في البداية الحالة الأولى للمسألة.
- **Close**: تضم عقد البيان التي جرت معالجتها سابقاً.

تُطبق إجرائياتُ البحث العمياء المؤثراتِ على العقد دون استعمال أي معرفة خاصة بمجال المسألة (غير تلك المعرفة حول شرعية الأفعال). ويمكن لهذه الإجرائيات أن تعتمد إحدى طريقتين:

- البحث عرضاً_أولاً،
- البحث عمقاً_ أولاً.

البحث عرضاً_أولاً